



UNIVERSIDAD NACIONAL DE COLOMBIA

SEDE MEDELLÍN

**DISEÑO DE UN ESQUEMA DE COORDINACIÓN DE
COMPORTAMIENTOS PARA LA NAVEGACIÓN DE UNA
PLATAFORMA ROBÓTICA MÓVIL**

ORLANDO ZAPATA CORTES, Ing.

**UNIVERSIDAD NACIONAL DE COLOMBIA
FACULTAD DE MINAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN Y DE LA DECISIÓN
MAESTRÍA EN INGENIERIA DE SISTEMAS
MEDELLÍN
2013**

**DISEÑO DE UN ESQUEMA DE COORDINACIÓN DE
COMPORTAMIENTOS PARA LA NAVEGACIÓN DE UNA
PLATAFORMA ROBÓTICA MÓVIL**

ORLANDO ZAPATA CORTES, Ing.

Tesis de investigación presentada como requisito para optar al título de Magister
en Ingeniería de Sistemas

Director:

JOVANI ALBERTO JIMÉNEZ BUILES, Ph. D.
Doctor en Ingeniería – Sistemas e Informática
Universidad Nacional de Colombia

Co-Director

GUSTAVO ALONSO ACOSTA AMAYA, M. Sc.
Magister en Ingeniería – Sistemas e Informática
Universidad Nacional de Colombia

LINEAS DE INVESTIGACIÓN:

Robótica móvil, inteligencia artificial, control inteligente, mecatrónica

GRUPO DE INVESTIGACIÓN:

Inteligencia Artificial en Educación

**UNIVERSIDAD NACIONAL DE COLOMBIA
FACULTAD DE MINAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN Y DE LA DECISIÓN
MAESTRÍA EN INGENIERIA DE SISTEMAS
MEDELLÍN
2013**

DEDICATORIA

Esta dedicatoria va dirigida a mi familia Orlando, Mariela, Julián y Ana por el apoyo en todo momento y agradecerles por su ayuda incondicional en la búsqueda del cumplimiento de este objetivo. También agradezco a Luisa por acompañarme por tanto tiempo en mi vida, dándome siempre una voz de aliento para seguir adelante. Dedico también este trabajo al profesor Gustavo Alonso Acosta Amaya que de buena fe fue un gran contribuyente durante este proceso como orientador, consejero en el ámbito académico y el cual lo considero excelente como docente y persona.

AGRADECIMIENTOS

El autor de esta investigación expresa con gratitud agradecimientos a Jovani Alberto Jiménez Builes, Ph. D. Director de esta investigación, por su orientación y asesoría en la culminación de este trabajo de investigación. A Él muchas gracias por su amistad brindada y el grato acompañamiento durante este proceso de formación en la Universidad Nacional de Colombia. También agradezco a Gustavo Alonso Acosta Amaya, M. Sc. por su amplia colaboración, Transferencia de su conocimiento, apoyo y su motivación constante durante la ejecución de esta investigación.

Resumen

En esta tesis de maestría se diseñó un esquema de coordinación de comportamiento para una plataforma robótica móvil. Inicialmente se diseñó un conjunto de comportamiento el cual está constituido por el seguimiento de pared derecha, seguimiento de pared izquierda y evasión de obstáculos. Como técnica de diseño, se utilizó lógica difusa para integrar el conocimiento cuantitativo y cualitativo de los expertos en los comportamientos del robot. Consecuentemente se diseñó un esquema de coordinación de comportamientos donde se propone una estrategia de fusión mediante una ponderación contextual dinámica difusa asociada a las regiones que el robot transita en su navegación. Por último los resultados de la implementación del esquema propuesto se confrontan con los resultados obtenidos de la implementación de un esquema convencional utilizado para la coordinación de comportamientos en robots móviles.

Palabras claves: Robótica móvil, lógica difusa, coordinación de comportamiento, control basado en comportamientos, control inteligente, inteligencia artificial.

Abstract

This thesis presents a behavior coordination scheme for a mobile robotic platform was designed. Initially a set of behavior which is made by right wall tracking, left wall tracking and obstacle avoidance was designed. As a design technique, fuzzy logic was used to integrate quantitative and qualitative expert knowledge in robot behaviors. Consequently a coordination of behaviors where fusion strategy is proposed using a fuzzy dynamic contextual weight associated to the regions that the robot travels in his navigation was designed. Finally the results of the implementation of the proposed scheme are compared with the results obtained from the implementation of a conventional scheme used for the coordination of behaviors on mobile robots.

Keywords: Mobile Robotics, fuzzy logic, coordination behavior, behavior-based control, intelligent control, artificial intelligence.

Tabla de contenido

Lista de figuras	11
Lista de tablas	15
Abreviaturas.....	16
Capítulo 1. Preliminares	17
1.1. INTRODUCCIÓN	17
1.2. PLANTEAMIENTO DE PROBLEMA.....	19
1.3. JUSTIFICACIÓN.....	20
1.4. PREGUNTA DE INVESTIGACIÓN	21
1.5. HIPÓTESIS	21
1.6. OBJETIVOS	21
1.6.1. Objetivo general	21
1.6.2. Objetivos específicos.....	21
1.7. MARCO TEÓRICO	22
1.7.1. Robótica.....	22
1.7.2. Sensórica en robots móviles.....	25
1.7.2.1. Sensórica exteroceptiva	26
1.7.2.1.1. Sensor infrarrojo.....	26
1.7.2.1.2. Sensores de ultrasonido	27
1.7.2.2. Sensórica propioceptiva	28
1.7.2.2.1. Sensor odométrico	28
1.8. ESTADO DEL ARTE	29
1.8.1. Control basado en comportamientos	32
1.9. METODOLOGÍA PROPUESTA.....	41
1.10. CONCLUSIONES DEL CAPÍTULO.....	43
Capítulo 2. Descripción de la Plataforma Robótica	45

2.1.	MÓDULO DE COMUNICACIONES	46
2.2.	MÓDULO DE PERCEPCIÓN	48
2.2.1.	Características del sonar SRF02.....	50
2.2.2.	Fenómenos asociados.	52
2.3.	MÓDULO DE CONTROL.....	52
2.4.	MÓDULO DE ACTUADORES	55
2.5.	CONCLUSIONES DEL CAPÍTULO	59
Capítulo 3.....		61
Objetivo 1: Comportamientos en Robots Móviles.....		61
3.1.	INVENTARIO DE COMPORTAMIENTOS	61
3.1.1.	Comportamiento 1: Seguimiento de Paredes (Wall Following Behavior)	61
3.1.2.	Comportamiento 2: Evasión de Obstáculos (Obstacle Avoidance Behavior)...	63
3.1.3.	Comportamiento 3: Búsqueda de Objetivos (Goal Seeking Behavior)	65
3.1.4.	Comportamiento 4: Evasión de Estancamiento (Deadlock Disarming Behavior)	66
3.1.5.	Comportamiento 5: Comportamiento de Emergencia (Emergency Behavior)	67
3.1.6.	Comportamiento 6: Vagabundeo (Wandering).....	68
3.1.7.	Comportamiento 7: Bloqueo de Motores (Locked-Rotor Detection Behavior) .	68
3.2.	CONCLUSIONES DEL CAPITULO	69
Capítulo 4.....		70
Objetivo 2: Diseño de Comportamientos		70
4.1.	CONTROL POR DECISIONES ON - OFF	72
4.1.1.	Pruebas.....	75
4.1.1.1.	Prueba 1.....	76
4.1.1.2.	Resultados prueba 1	78
4.1.1.3.	Prueba 2.....	82
4.1.1.4.	Resultados prueba 2.....	84
4.1.2.	Conclusiones de las Pruebas	87
4.1.3.	Premisa.....	87
4.2.	DISEÑO DE COMPORTAMIENTO DE SEGUIMIENTO DE PARED DERECHA	87
4.2.1.	Diseño.....	89

4.2.2.	Controlador difuso.....	92
4.2.2.1.	Caracterización de variables lingüísticas para el comportamiento seguimiento de pared derecho.....	92
4.2.2.2.	Tabla FAM	95
4.2.2.3.	Simulación del comportamiento.....	96
4.3.	DISEÑO DE COMPORTAMIENTO EVASIÓN DE OBSTÁCULOS	97
4.3.1.	Diseño	99
4.3.2.	Controlador difuso.....	104
4.3.2.1.	Caracterización de variables lingüísticas para el comportamiento evasión de obstáculos.....	104
4.3.2.2.	Tabla FAM	107
4.3.2.3.	Simulación del comportamiento.....	108
4.4.	DISEÑO DE COMPORTAMIENTO DE SEGUIMIENTO DE PARED IZQUIERDA 111	
4.4.1.	Diseño	111
4.4.2.	Controlador difuso.....	115
4.4.2.1.	Caracterización de variables lingüísticas para el comportamiento seguimiento de pared izquierda.....	115
4.4.2.2.	Tabla FAM	117
4.4.2.3.	Simulación del comportamiento.....	118
5.1.	PRUEBAS Y RESULTADOS	119
5.1.1.	Metodología.....	119
5.1.2.	Seguimiento de pared derecha.....	120
5.1.3.	Seguimiento de pared izquierda.....	122
5.1.4.	Evasión de obstáculos.....	124
5.2.	CONCLUSIONES DEL CAPÍTULO	127
Capítulo 5.....		129
Objetivo 3: Esquema de coordinación de comportamientos.....		129
5.1.	DISEÑO DEL ESQUEMA DE COORDINACIÓN	130
5.2.	FUSIÓN POR PONDERACIÓN CONTEXTUAL DINÁMICA BASADA EN REGIONES (FPCDBR).....	133
5.3.	DISEÑO DEL SISTEMA DIFUSO DE PONDERACIÓN	135
5.3.1.	Tabla FAM	136

5.3.2. Reglas de inferencia	136
5.4. CONCLUSIONES DEL CAPÍTULO	136
Capítulo 6.....	138
Objetivo 4: Evaluación de esquema de coordinación de comportamientos. (Pruebas y resultados)	138
6.1. ESQUEMA DE SUBSUNCIÓN PARA EL ROBOT <i>Carlitos</i>	139
6.1.2. Pruebas experimentales y resultados: esquema de subsunción.....	140
6.1.2.1. Seguimiento de pared derecha y evasión de obstáculos	141
6.1.2.2. Seguimiento de pared izquierda y evasión de obstáculos	145
6.1.3. Conclusiones del uso del esquema de subsunción.....	148
6.2. CONFRONTACIÓN DE LOS ESQUEMAS DE CONTROL	148
6.3. CONCLUSIONES DE CAPÍTULO.....	154
Conclusiones.....	155
Trabajo a futuro	156
Bibliografía.....	157
Anexos A. Diagramas de Flujo de programas y subrutinas	163
Anexos B. Interfaz de adquisición de datos.....	164

Lista de figuras

Figura 1.1. Robot Atlas (2013), Boston Dynamics	22
Figura 1.2. Robot Petman (2013), Boston Dynamics	23
Figura 1.3. Robot KR 100 TITAN.	23
Figura 1.4. Robot Roomba 780.	24
Figura 1.5. Big Dog, Boston Dynamics.	24
Figura 1.6. Minerva, the Smithsonian's National Museum of American History.....	25
Figura 1.7. Sensor óptico ("Incon - Produtos Eletrônicos," 2013).	26
Figura 1.8. Sensor de ultrasonido emisor y receptor (Super Robotica, 2013).	27
Figura 1.9. Encoder de cuadratura de efecto Hall (Pololu, 2013).	28
Figura 1.10. Arquitectura de cuatro niveles del Robot Walling.(Acosta Amaya, 2010, p. 72)	30
Figura 1.11. Resultados obtenidos en el mapeo por el robot walling.(Acosta Amaya, 2010, p. 106)	30
Figura 1.12. Robot Carlitos. (Zapata & Herrera, 2011).....	31
Figura 1.13. Resultados obtenidos en el mapeo por el robot Carlitos.(Zapata & Herrera, 2011, p. 106)	32
Figura 1.14. Arquitecturas de navegación. a) Deliverativa, b) Reactiva	34
Figura 1.15. Arquitectura de subsunción (Brooks, 1986).....	35
Figura 1.16. Tareas en paralelo (Brooks, 1986).....	35
Figura 1.17. Clases de mecanismos de coordinación (MacKenzie et al., 1997).....	36
Figura 1.18. Mecanismos de coordinación de comportamiento (Pirjanian, 1999).	36
Figura 1.19. Esquema para el método de campos potenciales (Wicaksono et al., 2009) .	38
Figura 1.20. Esquema de coordinación de comportamientos basado en lógica difusa (Wicaksono et al., 2009).....	39
Figura 1.21. Esquema de coordinación de comportamientos neuro-difuso (Song & Lin, 2006)	39
Figura 1.22. a) Arquitectura reactiva b) arquitectura deliberativa (Acosta Amaya, 2010)..	41
Figura 1.23. Arquitectura híbrida (deliberativa-reativa) (Acosta Amaya, 2010).	41
Figura 2.1. Plataforma robótica <i>CARLITOS</i> (acrónimo de Cartographic Intelligent Robot for SLAM.)	45
Figura 2.2. Enlace de comunicación Xbee.	46
Figura 2.3. a) Xbee Explorer Dongle; b) comunicación Xbee-USB.....	47
Figura 2.4. Estructura general de comunicación Ordenado-Xbee-Microcontrolador.....	48
Figura 2.5. Interconexión los sonares.	49
Figura 2.6. Anillo de sonares.	49
Figura 2.7. Modo de trabajo de UART(Serial) a I2C.....	50
Figura 2.8. Diagrama de pines.....	51

Figura 2.9. Lóbulo de radiación.	52
Figura 2.10. Módulo de control.	53
Figura 2.11. Conexiones del microcontrolador.	54
Figura 2.12. Especificaciones globales del microcontrolador (Freescale, 2013).	55
Figura 2.13. Módulo de actuadores.	56
Figura 2.14. Esquemático configuración puente en H - L298.	57
Figura 2.15. Accionamiento de motores.	58
Figura 2.16. Cinemática en robots diferenciales(Navarro et al., 2007).	59
Figura 3.1. Esquema de seguimiento de paredes por separado.	62
Figura 3.2. Detección de paredes en simultáneo.	62
Figura 3.3. Detección en simultáneo (Song & Lin, 2006).	63
Figura 3.4. Ejemplo de esquema de evasión de obstáculos frontales (Peng & Yumei, 2009).	64
Figura 3.5. Simulación de un robot evadiendo un obstáculo(Shim et al., 2011)	65
Figura 3.6. Goal Seeking Behavior (Song & Lin, 2006).	65
Figura 3.7. Deadlock disarming behavior (Dongshu et al., 2011).	66
Figura 3.8. Emergency Behavior(EB).	67
Figura 3.9. Wandering (Dongshu et al., 2011).	68
Figura 4.1. Proceso utilizado en el diseño de los comportamientos utilizando lógica difusa.	71
Figura 4.2. Zonas consideradas por el algoritmo de control.	73
Figura 4.3. Diagrama de flujo – Algoritmo de control.	74
Figura 4.4. Escenario real.	76
Figura 4.5. Escenario 1 de pruebas.	77
Figura 4.6. Inicio de recorrido, T(1-4)“Trayectorias”.	78
Figura 4.7. Oscilaciones iniciales.	79
Figura 4.8. Sobreimpulso.	79
Figura 4.9. Referencia.	80
Figura 4.10. Mediciones de sobreimpulsos.	80
Figura 4.11. Error de orientación.	81
Figura 4.12. Escenario 2 de pruebas.	83
Figura 4.13. Inicio de recorrido, T(1-4)“Trayectorias” y oscilaciones iniciales.	84
Figura 4.14. Mediciones de sobreimpulsos.	84
Figura 4.15. Error de orientación.	85
Figura 4.16. Seguimiento de pared derecha	88
Figura 4.17. Análisis del error y la derivada del error de medida de S4.	90
Figura 4.18. Diagrama de control <i>fuzzy</i> seguimiento de pared derecha.	92
Figura 4.19. Estructura de control del seguimiento de pared derecha.	92

Figura 4.20. Partición de variables lingüísticas de entrada en conjuntos difusos, salida PWM y superficie de control para el comportamiento de seguimiento de pared derecha.	96
Figura 4.21. Sensores frontales.	98
Figura 4.22. Detección obstáculo costado frontal izquierdo.	100
Figura 4.23. Detección de obstáculos frontales derechos. b) Detección de obstáculos frontales.	101
Figura 4.24. Diagrama de control <i>fuzzy</i> evasión de obstáculos.	103
Figura 4.25. Estructura de control de evasión de obstáculos.	104
Figura 4.26. Tabla FAM evasión de obstáculos.	107
Figura 4.27. Partición de variables lingüísticas de entrada en conjuntos difusos, salida PWM y PWMI	108
Figura 4.28. Relación de las variables de entrada eS1, efl, efD respecto a las salidas PWM y PWMI	109
Figura 4.29. Análisis del error y la derivada del error de medida de S10.	113
Figura 4.30. Diagrama de control <i>fuzzy</i> seguimiento de pared izquierda.	114
Figura 4.31. Estructura de control del seguimiento de pared izquierda.	115
Figura 4.32. Partición de variables lingüísticas de entrada en conjuntos difusos, salida PWM y superficie de control para el comportamiento de seguimiento de pared izquierda.	118
Figura 4.33. Navegación - Seguimiento de pared derecha.	120
Figura 4.34. Datos suministrados por el robot.	121
Figura 4.35. Navegación - Seguimiento de pared izquierda.	122
Figura 4.36. Datos suministrados por el robot.	124
Figura 4.37. Escenario de navegación propuesto para pruebas de evasión de obstáculos.	125
Figura 4.38. Datos de detección de objetos frontales izquierdos.	126
Figura 4.39. . Datos de detección de objetos frontales derechos.	126
Figura 4.40. Datos de detección de objetos frontales directos.	126
Figura 5.1. Simplificación de los sistemas de control	130
Figura 5.2. Simplificación de los diagramas a) seguimiento de pared izquierda b) evasión de obstáculos.	131
Figura 5.3. Esquema de control - Fusión por ponderación contextual dinámica difusa basada en regiones (FPCDBR).	132
Figura 5.4. Discriminación de entorno de navegación en regiones.	134
Figura 5.5. Simulación de sistema difuso para los cálculos de los factores de ponderación para las recomendaciones del motor derecho.	135
Figura 6.1. Subsunción(Paolo Pirjanian, 1999).	138
Figura 6.2. Arquitectura de Subsunción para el robot Carlitos.	139
Figura 6.3. Arquitectura de Subsunción para el robot Carlitos de manera explícita.	139

Figura 6.4. Escenario de navegación.....	141
Figura 6.5. a) Seguimiento de pared derecha-evasión de obstáculos, b) Error de llegada, c) Evasión de esquina derecha.	142
Figura 6.6. Adquisición de datos - Seguimiento de pared derecha y evasión de obstáculos.	143
Figura 6.7. Error frontal derecho.....	144
Figura 6.8. Error frontal.	144
Figura 6.9. a) Evasión esquina izquierda, b) Error de llegada, c) Seguimiento de pared izquierda-evasión de obstáculos	145
Figura 6.10. Adquisición de datos - Seguimiento de pared Izquierda y evasión de obstáculos.....	146
Figura 6.11. Error frontal izquierdo.....	147
Figura 6.12. Error frontal	147
Figura 6.13. Confrontación de esquemas de coordinación de comportamientos.	149
Figura 6.14. Región 1: Seguimiento de pared derecha.....	150
Figura 6.15. Región 2: Evasión de obstáculo.....	151
Figura 6.16. a) Región 3: Seguimiento de pared derecha; b) Error de llegada.....	152
Figura 6.17. SUBSUNCIÓN VS FPCDBR	153
 Figura A1. Diagrama de flujo programa principal.....	163
Figura A2. Rutina de atención a interrupción por recepción de datos del puerto serial en el robot <i>Carlitos</i>	163
Figura A3. Rutina de atención a interrupción por desbordamiento del reloj de tiempo real.	163
 Figura B 1. Interfaz diseñada para adquirir los datos de los sensores.	164
Figura B 2. Diagrama de flujo del software de adquisición de datos.	164

Lista de tablas

Tabla 1. Especificaciones técnicas sonar SRF02.....	51
Tabla 2. Variables de entrada y salida propuestas para el conjunto de comportamientos diseñados.....	71
Tabla 3. Ejecución de los recorridos prueba 1	77
Tabla 4. Mediciones de sobreimpulso.....	81
Tabla 5. Error de orientación.	82
Tabla 6. Ejecución de los recorridos prueba 2.	83
Tabla 7. Mediciones de sobreimpulso.....	85
Tabla 8. Error de orientación.	86
Tabla 9. Análisis del comportamiento-Seguimiento de pared derecha.....	91
Tabla 10. Caracterización variable de error seguimiento de pared derecha.....	93
Tabla 11. Caracterización variable derivada del error del seguimiento de pared derecha.....	93
Tabla 12. Caracterización variable de salida de DPWMD.	94
Tabla 13. Tabla FAM seguimiento de pared derecha.....	95
Tabla 14. Caracterización variable de error frontal izquierdo.	105
Tabla 15. Caracterización variable de error frontal derecho.	105
Tabla 16. Caracterización variable de error S1.....	106
Tabla 17. Caracterización variable de salida de DPWMED.....	106
Tabla 18. Análisis del comportamiento-Seguimiento de pared izquierda.....	114
Tabla 19. Caracterización variable de error seguimiento de pared izquierda.	115
Tabla 20. Caracterización variable derivada del error del seguimiento de pared izquierda.	116
Tabla 21. Caracterización variable de salida de DPWMEI.....	116
Tabla 22. Tabla FAM seguimiento de pared izquierda.....	117
Tabla 23. Datos de las navegaciones.	121
Tabla 24. Datos de las navegaciones.	123
Tabla 25. Tabla FAM asociada al contexto para el cálculo de los factores de ponderación difusos.....	136

Abreviaturas

PWM	Pulse Width Modulation
DPWMDW	Delta de PWM Derecho del seguimiento de pared (Wall) derecha
DPWMIW	Delta de PWM Izquierdo del seguimiento de pared (Wall) izquierda
DPWMED	Delta de PWM Evasión Derecho
DPWMEI	Delta de PWM Evasión Izquierdo
DPWMDF	Delta de PWM Derecho Final
DPWMIF	Delta de PWM Izquierdo Final
PD:	Factor de Ponderación dinámico difuso Derecho
PI	Factor de Ponderación dinámico difuso Izquierdo
PEOD	Factor de Ponderación dinámico difuso Evasión Obstáculos Derecha
PEOI	Factor de Ponderación dinámico difuso Evasión Obstáculos Izquierda
RF	Radio frecuencia
USB	Universal Serial Bus
Tx	Transmisión
Rx	Recepción
UART	Universal Asynchronous Receiver-Transmitter
I2C	Inter-Integrated Circuit
DC	Duty Cycle
e	Error
eS	Error sensor
de	Derivada del error
deS	Derivada del error Sensor
eFD	Error frontal derecho
eFI	Error frontal izquierdo
SP	Set-Point
trapmf	Trapezoidal-shaped membership function
trimf	Triangular-shaped membership function
Pt	Prositivo
Zr	Zero
Nt	Negativo
NG	Negativo Grande
NP	Negativo Pequeño
PP	Positivo Pequeño
PG	Positivo Grande

Capítulo 1. Preliminares

1.1. INTRODUCCIÓN

Los avances tecnológicos en robótica móvil han ido generando desafíos cada vez más complejos con un sinnúmero de exigencias para los investigadores que frecuentan este campo de investigación. Tales requerimientos han marcado pautas en donde las técnicas para abordar estos problemas deben ser alterna y diferente a los métodos tradicionales. En robótica móvil el uso de técnicas de inteligencia artificial se ha posicionado de tal manera que se ha convertido en uno los métodos más atractivos y de preferencia por la comunidad científica que trabaja en robótica, por la gran flexibilidad que brinda en el diseño y la capacidad que tiene al integrar conocimiento cualitativo y cuantitativo de los expertos o personas con un conocimiento profundo en el ámbito de la robótica móviles.

La implementación de las técnicas de inteligencia artificial como las redes neuronales, lógica difusa, algoritmos genéticos, programación evolutiva, métodos heurísticos, entre otras, han generado un amplio panorama para dar soluciones de los problemas en robótica. Estas técnicas poseen tal atractivo que en ocasiones se recurre a métodos híbridos entre ellas para generar nuevos métodos de solución. Estas técnicas también han fortalecido y contribuido a los método tradicionales como el caso del control convencional en combinación con lógica difusa, dando esto como resultado un control difuso que entra en la categoría de control inteligente.

En robótica, el control de robots móviles de basa generalmente en dos métodos: el control basado en comportamiento y el control con planificación de trayectorias. El control basado en comportamiento como su nombre lo indica es la reacción por medio de aptitudes del robot frente a las situaciones que puede encontrar en su navegación. En contraste está el control con planificación de trayectorias donde el robot posee un conocimiento a priori del entorno y planea su navegación.

El control basado en comportamiento se basa en las múltiples aptitudes que puede tomar el robot frente al entorno en que se desenvuelve, estas aptitudes son llamadas comportamientos. El diseño de las arquitecturas de control basado en comportamiento en robots móviles tiene principalmente dos problemas: el diseño y coordinación. Los problemas de diseño de comportamientos están asociados a la inteligencia que debe tener la aptitud: cómo plasmar esa inteligencia en el comportamiento y qué sensorica utilizar en el comportamiento para que el robot interactúe satisfactoriamente con el ambiente. Por otra parte el problema de coordinación radica en cómo solucionar el conflicto cuando varios comportamientos tratan de tomar el control de los efectores en un determinado momento.

Como solución a los problemas presentados en el control basado en comportamientos se ha utilizado el control inteligente como alternativa de solución, en donde todo el conocimiento de este control se fusiona a los métodos de diseño y coordinación de comportamiento en robótica móvil teniéndose como resultante un control inteligente basado en comportamiento.

El propósito de esta investigación consistió en diseñar un esquema de coordinación de comportamientos para la navegación de una plataforma robótica móvil con la cual se busca contribuir a la solución de los problemas identificados en el control basado en comportamientos por medio del control inteligente.

La motivación en esta tesis de investigación es presentar y dar a conocer el potencial que se puede desarrollar con la integración del control inteligente al diseño y coordinación de comportamientos en robots dotados con dispositivos electrónicos de medianas prestaciones.

Para el cumplimiento de los objetivos de esta tesis de investigación se plantearon unos lineamientos de los cuales el primero de ellos, fue buscar entre el *background* existente de control basado en comportamiento, qué comportamientos son los más representativos y utilizados en robótica móvil que aseguren al autómatas una navegación satisfactoria en entornos interiores de trabajo y estáticos. Posterior a esto se seleccionó un conjunto de comportamientos que fueron diseñados con control

inteligente, simulados y chequeados en el robot *Calitos*. Después de la realimentación y los ajustes gracias a la experimentación exhaustiva realizada con los comportamientos, se procedió a realizar un esquema de coordinación que administre estos de manera adecuada. Por último se realizaron pruebas de chequeo de la integración de los comportamientos al esquema de coordinación formándose por completo el esquema de control general para la plataforma robótica *Carlitos*.

1.2. PLANTEAMIENTO DE PROBLEMA

El afán del ser humano por expandir sus tareas de exploración en terrenos donde sus limitaciones físicas le impiden acceso directo, ha generado la necesidad de adoptar sistemas robóticos inteligentes capaces de ejecutar tareas programadas para el cumplimiento de sus objetivos o labores de investigación. Las características de los sistemas robóticos utilizados en exploración son generadas indirectamente por los ambientes donde será expuesto el robot, dichos requerimientos se traducen en la incorporación de sensores adecuados para exploración, resistencias de los materiales del chasis, configuración de los efectores del robot (ruedas, ruedas tipo oruga, bípedos, etc.) y su capacidad de cómputo como eje principal y cerebro del agente robótico (Zapata & Herrera, 2011).

Las exigencias de exploración en la actualidad han originado la necesidad de incorporar sistemas robóticos más inteligentes y compactos, caso del cual no se excluye la exploración de entornos interiores con robots móviles de pequeñas dimensiones. Esta clase de robots ejecutan comportamientos programados los cuales deberán trabajar en conjunto, para el cumplimiento de sus tareas. Los problemas encontrados en esta área de la robótica móvil son el diseño de los comportamientos y su coordinación para apropiarse de los efectores finales (ruedas) en los contextos presentados por el ambiente (Adriansyah & H. M. Amin, 2005).

El diseño de un esquema de coordinación de comportamientos para navegación con robots móviles pequeños (Small wheeled mobile robots) mejoraría las capacidades de navegación en autómatas móviles con sensorica y capacidad limitada (Qing-yong et al., 2009).

1.3. JUSTIFICACIÓN

En el campo de la robótica en los últimos años, la comunidad científica se ha esmerado por construir prototipos robóticos cada vez más versátiles con características que mejoran el desempeño en las tareas asignadas para el autómata. Estos esfuerzos han llevado la robótica móvil a incurrir en diferentes campos como la domótica, la Industria, exploración de terrenos, la robótica militar, etc.

El éxito de la robótica móvil en las diferentes áreas ha venido siendo cada vez más notable y reconocido por la comunidad mundial, dando esto como resultados exigencias de sistemas robóticos cada vez más autónomos. Tales exigencias requieren el uso de técnicas de inteligencia artificial y mecanismos de coordinación que brinden al robot móvil suficiente independencia y agilidad.

El uso de técnicas y métodos inteligente se han convertido en un pilar fundamental para la construcción de comportamiento y esquemas de coordinación de comportamiento en robótica móvil, tal es el caso del éxito de algunas técnicas de inteligencia artificial como son los algoritmos genéticos, lógica difusa, redes neuronales, sistemas híbridos neuro-difusos y programación evolutiva, las cuales van en pro de soluciones más efectivas y eficientes en la programación de los robots móviles (Adriansyah & H. M. Amin, 2005).

Sin embargo siguen existiendo problemas asociados a las conductas que debería tomar el robot móvil de acuerdo al contexto presentado por el entorno y la coordinación de estos comportamientos. Por las razones presentadas anteriormente, para la navegación de robots móviles pequeños se diseñó un

esquema de coordinación de comportamientos como una alternativa más a la solución a los problemas mencionados anteriormente.

1.4. PREGUNTA DE INVESTIGACIÓN

¿Qué beneficios tendrá el diseño e implementación de un esquema de coordinación que mejore la administración de los comportamientos para una plataforma robótica móvil en entornos interiores cerrados y estáticos?

1.5. HIPÓTESIS

Un esquema de coordinación de comportamiento permitirá gestionar de manera eficiente el conjunto de conductas preestablecidas para el robot, evitando los conflictos que puedan surgir entre los diferentes comportamientos para ganar el control de los efectores. Así mismos permitirá obtener trayectorias suaves y continuas que mejorarán la navegación del vehículo autónomo en su entorno.

1.6. OBJETIVOS

1.6.1. Objetivo general

Diseñar un esquema de coordinación de comportamientos que permita la obtención de trayectorias suaves de navegación en entornos interiores de trabajo.

1.6.2. Objetivos específicos

- Caracterizar los comportamientos más representativos en el control de navegación de robots móviles en entornos interiores.
- Diseñar un conjunto básico de comportamientos que garantice la navegación segura del robot móvil en su entorno de trabajo.

- Implementar un esquema de coordinación que permita la integración de los comportamientos básicos establecidos.
- Evaluar el desempeño de la arquitectura de control integrando el esquema de coordinación de comportamientos.

1.7. MARCO TEÓRICO

1.7.1. Robótica

En la actualidad se fabrican diferentes sistemas robóticos. Estos robots están destinados a suplir tareas de tipo industrial, comercial, servicios al cliente, tareas militares, tareas de exploración submarina, guías, etc. La comunidad científica especializada en esta área siempre se ha interesado primordialmente en el diseño y construcción de robots que emulen el ser humano tanto en lo físico como en su capacidad de razonar. De allí surge la relación que la robótica será el futuro de la humanidad.

Entre las aproximaciones más importantes que ha tenido la robótica hacia el ser humano se destacan los robots antropomórficos Atlas y Pecman (ver figuras 1.1 y 1.2) desarrollados por la compañía de ingeniería Boston Dynamics, compañía especializada en la construcción robots y software para la simulación de la dinámica humana.



Figura 1.1. Robot Atlas (2013), Boston Dynamics

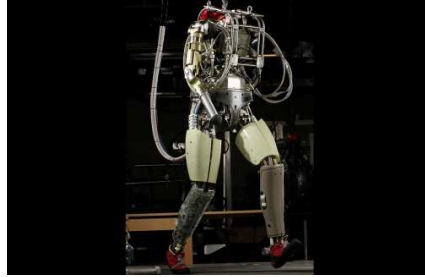


Figura 1.2. Robot Petman (2013), Boston Dynamics

Dentro de los robots industriales existe una alta gama de autómatas los cuales son utilizados en labores arduas y repetitivas tales como soldadura, transporte de suministro en las empresas, pintura y ensamble entre otras. Estos sistemas robóticos de la industria también se diferencian según las características requeridas por la empresa pues estos pueden trabajar de forma autónoma durante largas jornadas o simplemente ser tele-operados por un operario en tareas de excesivo cuidado.

Entre los robots industriales se destaca el robot '*KR 100 TITAN*',(ver figura 1.3) el cual es el robot manipulador de 6 grados de libertad más grande y fuerte encontrado en el mercado con una capacidad de carga de hasta 1000kg, desarrollados por la empresa alemana *kuka-robotics*. Este robots es configurado con trayectorias en un espacio 3D las cuales se traducen en la ejecución de una tarea de ensamble, soldadura, pintura o transporte de carga según su efector final.



Figura 1.3. Robot KR 100 TITAN.

Otros robots tales como los prestadores de servicio, se encuentran los autómatas desarrollados por la empresa iRobot especializados en labores domésticas como aspirar el piso de una casa (Robot Roomba 780) (ver figura 1.4). El cual cuenta con una interfaz táctil para comunicarse con el usuario, con la cual se puede programar el autómata para que inicie sus labores de limpieza automáticamente. En las características principales de Roomba resaltan su eficiente software y sensores para la navegación de limpieza de los suelos de forma efectiva y su sistema Cliff Detect que evita que el robot caiga por escaleras o desniveles.



Figura 1.4. Robot Roomba 780.

Como ejemplo en los robots militares de última generación (2013) se encuentra el robot '*Big Dog*' (ver figura 1.5) construido por *Boston Dynamics* el cual es considerado el macho alfa de los robots construidos por ésta empresa, ya que cuenta con las habilidades de correr, caminar, escalar obstáculos y llevar cargas pesadas.



Figura 1.5. Big Dog, Boston Dynamics.

Y entre los robots guías más representativos existe el autómata '*Minerva*', exhibido en el museo Smithsonian (Washington) capaz de interactuar con personas. Está dotado con un módulo de sensores y una arquitectura de software capaz de dar autonomía suficiente en la navegación de entornos impredecibles y dinámicos. Este robot tiene la facultad de mapear, navegar, realizar mapas de textura del techo del museo, brinda sus coordenadas para su localización y planificación de rutas generadas por él mismo (Thrun et al., 1999).



Figura 1.6. Minerva, the Smithsonian's National Museum of American History.

1.7.2. Sensórica en robots móviles

En la construcción de robots en general se necesitan componentes tanto de software (lenguajes y técnicas de programación) como hardware (actuadores, sensores, microcontroladores, procesadores, sistemas de comunicaciones programables, etc.). Dentro de los componentes de hardware, los sensores en particular tienen mayor importancia ya que estos dispositivos se establecen como elementos principales que interactúan con el ambiente.

Es indispensable entonces dotar los robots con dispositivos capaces de percibir señales tanto del ambiente que los rodea (objetos reflectores, paredes, obstáculos

en general) y el estado en que se encuentran los actuadores en un momento determinado (velocidad promedio de motores, distancia recorrida por el robot, posición y orientación).

Uno de los objetivos principales de la incorporación de sensores en robots móviles es servir como soporte para el cumplimiento de las tareas del robot, dando la suficiente capacidad de interactuar con el ambiente para evitar situaciones como desniveles, pasillo sin salida, caídas, sillas, columnas, etc. Todo lo que pueda presenciar como obstáculo en un ambiente común. Los sensores que se han constituido como los más utilizados en el ámbito de la robótica móvil han sido los sensores ópticos infrarrojos y de ultrasonido.

1.7.2.1. Sensórica exteroceptiva

1.7.2.1.1. Sensor infrarrojo.

Los sensores infrarrojos tienen una alta precisión, sus medidas son menos sensibles en condiciones extremas, que en algunas ocasiones presenta el entorno, tales como diferenciales de presión y temperaturas variables. En la gama de sensores infrarrojos se pueden encontrar dispositivos de bajo o alto costo, pero generalmente un sensor infrarrojo de buena precisión conlleva un alto costo con respecto a otras alternativas que pueden dar resultados similares a bajo costo como el ultrasonido (Benet et al., 1998). En la figura 1.7 se puede ver la representación física de un sensor óptico.



Figura 1.7. Sensor óptico (“Incon - Produtos Eletrônicos,” 2013).

1.7.2.1.2. Sensores de ultrasonido.

En contraste a los sensores infrarrojos, se emplean los sensores de ultrasonido los cuales sus precios son significativamente bajos frente a los ópticos. Presentan una correlación Calidad-Precio. El uso es más simple y en algunos casos se encuentran integrados el emisor y receptor en un mismo dispositivo (Benet et al., 1998)(Navarro et al., 2004). En la figura 1.8 muestra los sensores de ultrasonido comúnmente utilizados.



Figura 1.8. Sensor de ultrasonido emisor y receptor (Super Robotica, 2013).

Los sensores de ultrasonido son versátiles incurriendo en muchas aplicaciones como medidores de nivel, velocidad, detección de objetos, proximidad de objetos, localización de objetos entre otras. Por tal motivo han sido ampliamente utilizados en robótica móvil por su amplia aplicabilidad, buena capacidad de trabajo y procesamiento de la información, otorgando altas velocidad de respuesta hacia la detección de objetos (Liu et al., 2011; Llata et al., 2002; Oria & Gonzalez, 1993; Sarabia et al., 1998). Estos dispositivos para su medición cuentan con un lóbulo de radiación o zona de detección, característica dada por el fabricante, la cual varía de acuerdo a la referencia del dispositivo.

Las desventajas que presentan estos sensores son de alto cuidado puesto que esto puede llegar a medidas erróneas. Entre los problemas más comunes en estos dispositivos se tiene la reflexión especular (reflexiones múltiples de la onda), absorción y atenuación del medio, sensible a ruidos e interferencias, sensibilidad a la temperatura y la humedad, y pobre resolución (G. Liu et al., 2011; D. Navarro et al., 2004).

1.7.2.2. Sensórica propioceptiva

1.7.2.2.1. Sensor odométrico

Como se explicó anteriormente los sensores ayudan al robot a percibir objetos que lo rodean, pero también existe la necesidad de percibir como está el sistema internamente. Esta clase de dispositivos recibe el nombre de sensórica propioceptiva, estos dispositivos generan señales que por medio de su análisis pueden dar un estado propio del robot en un instante deseado. Entre los sensores propioceptivos existen los sensores odométricos los cuales son *encoders* usualmente incrementales que sirven para estimar la posición actual con respecto a la posición inicial del robot y su orientación (Benet et al., 1998).

Los *encoders* usualmente se encuentran acoplados a las ruedas del robot y su función principal es dar información para estimar la posición del robot. En el mercado se puede encontrar básicamente dos tipos de *encoders*: absolutos e incrementales, pero también existen otras alternativas tales como *encoders* ópticos, lineales, de cuadratura que varían según su configuración, como por ejemplo *encoders* de cuadratura de efecto Hall. En la figura 1.9 muestra un *encoder* de cuadratura de efecto Hall.



Figura 1.9. Encoder de cuadratura de efecto Hall (Pololu, 2013).

1.8. ESTADO DEL ARTE

A nivel local son relevantes los siguientes trabajos en el área de robótica móvil para el mapeo y navegación.

- En su tesis de maestría “Ambiente Multi-Agente Robótico para la Navegación Colaborativa en Escenarios Estructurados” Gustavo Acosta Amaya presenta la elaboración de un sistema el cual tuvo como objetivo diseñar un Ambiente Multi-Agente Robótico para la navegación colaborativa en escenarios estructurados.

Como estructura física en la investigación se realizó la construcción de una plataforma robótica móvil (**Walling**) (ver figura 1.10) dotada de sensores y actuadores siendo está el agente encargado de la recolección de datos del entorno. También se desarrolló otro agente el cual fue el algoritmo codificado en MatLab (**Magallanes**) para la navegación, captura de datos y procesamiento de señales suministradas por el robot *walling*. Éste agente fue elaborado con un gran complejidad siendo capaz de dar al robot el rol de navegación, tomar decisiones frente a obstáculos, permanecer a una distancia constante hacia una pared en su recorrido, y por último se integró al sistema un algoritmo de mapeo (**Vespucci**) que realizó las representaciones computacionales de los entornos a los cuales fue expuesto el sistema (Acosta Amaya, 2010).

Los resultados arrojados en esta investigación se pueden observar en la figura 1.11, la cual presenta la aproximación digital de un entorno explorado por el robot *Walling*.

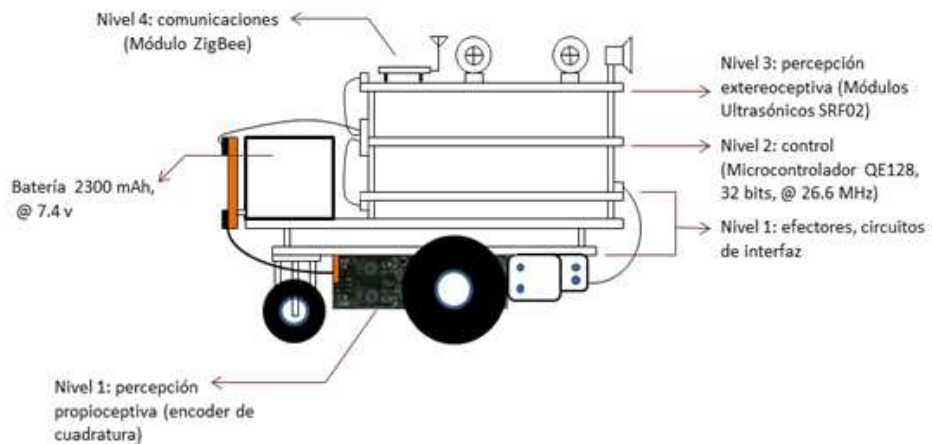


Figura 1.10. Arquitectura de cuatro niveles del Robot Walling.(Acosta Amaya, 2010, p. 72)



Figura 1.11. Resultados obtenidos en el mapeo por el robot Walling.(Acosta Amaya, 2010, p. 106)

En la conceptualización y construcción de plataformas robóticas móviles hasta el momento no se había tenido un registro de guía para el diseño de estos sistemas, por tal motivo se implementó como herramienta para suplir esta necesidad la metodología MAD-Smart, la cual está orientada hacia el análisis y diseño de Sistemas Multi-Agentes Robóticos, MAD-Smart tiene como objetivo describir las características de software y hardware del prototipo utilizando plantillas estándar como metodología para determinar los requerimiento (Jiménez, J. et al., 2007).

- **Carlitos:** Acrónimo de **Cartographic Intelligent Robot for SLAM**. Plataforma robótica para el mapeo y navegación de entornos interiores. Este robot está constituido por 3 módulos como estructura física (ver figura 1.12).



Figura 1.12. Robot Carlitos. (Zapata & Herrera, 2011)

El modulo uno consta de un anillo de sonares, 12 sensores de ultrasonido se encuentran posicionados en los lados de un dodecágono, posición diseñada para reducir las zonas muertas de detección entre sensores. Además éste módulo también está conformado por un dispositivo 'Xbee' capaz de transmitir datos a una distancia de 1.5 Km con respecto al dispositivo receptor.

El robot cuenta con un microcontrolador de 32 bits (*Coldfire V1 MCF51QE128*) Freescale Inc en su módulo de control, el cual es versátil y capaz de manejar salidas y entradas tanto análogas como digitales, PWM, conversores análogo-digitales, *timer*, comparadores análogos, entre otros. La programación de *Carlitos* se realizó bajo lenguaje C, lenguaje que sirvió como medio para programar los comportamientos de evasión de obstáculo frontal y seguimiento de pared derecha codificados en 2 set de instrucciones *IF-THEN* extraídas del análisis con lógica difusa. Éste módulo también cuenta con una brújula digital la cual brinda la orientación del robot con respecto al norte magnético terrestre, la cual sirve como ayuda en la construcción del mapa realizado por el autómata.

El tercer módulo de la estructura de *Carlitos* fue denominada módulo de actuadores, el cual está compuesto por dos motores en configuración diferencial los cuales poseen *encoders* de efecto Hall, como sensor propioceptivo y un puente en H(L295) que sirve como *driver* de control de los motores (Zapata & Herrera, 2011). En la figura 1.13 se presenta la aproximación geométrica del entorno explorado por el robot *Carlitos*.

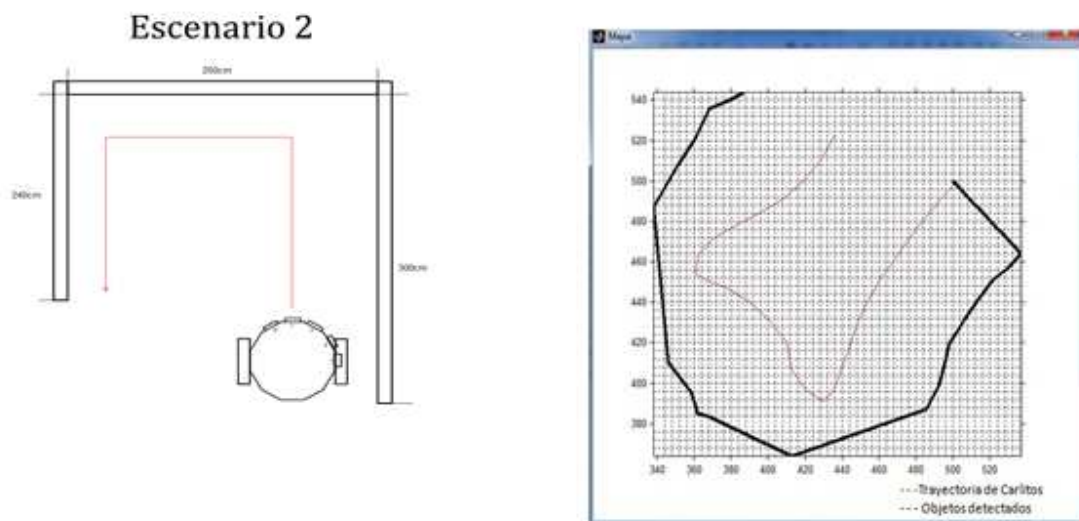


Figura 1.13. Resultados obtenidos en el mapeo por el robot Carlitos.(Zapata & Herrera, 2011, p. 106)

1.8.1. Control basado en comportamientos

Un robot móvil para navegar en un ambiente dinámico o estático utiliza comúnmente técnicas de control que aseguren un buen manejo de los efectores del robot con el fin de describir trayectorias fiables y evitando a toda costa una colisión con algún elemento que pueda presentar el entorno en el cual se desenvuelve.

Las técnicas de control utilizadas pueden ser tanto convencionales como inteligentes, las cuales tiene algunas ventajas y desventajas una con respecto a la otra. El control convencional en algunas ocasiones no puede ejercer acciones de control en límites donde el control inteligente lo hace con facilidad debido a la

flexibilidad que puede presentar en su diseño. Otra diferencia es que en la mayoría de los casos en el control convencional se debe conocer el modelo del sistema mientras que el control inteligente en algunas se puede trabajar sin dicho modelo o inclusive ser ignorado como es el caso del control basada en comportamientos (Qing-yong et al., 2009). Por las razones mencionadas anteriormente, el control inteligente se considera como una herramienta fácil de utilizar e implementar.

En el campo de robótica móvil las técnicas de control inteligente se han convertido en un mecanismo por el cual se pueden diseñar sistemas de control más flexibles con mayor robustez. Particularmente el control basado en comportamientos para robots móviles, el cual ha venido consolidado como la técnica más usada en la última década y ha tomado gran popularidad entre la comunidad científica que trabaja en este campo de la robótica móvil (Adriansyah & H. M. Amin, 2005; Arkin, 1998; Thongchai et al., 2000).

Según Arkin un comportamiento se define como:

“La reacción a un estímulo”(Ronald C. Arkin, 1998, p. 66)

Se infiere entonces que un sistema de control basado en comportamientos es una estructura de control basada en una serie de conductas que puede tomar el robot en una determinada situación de acuerdo a los estímulos que presenta el ambiente de navegación (Paredes, objetos, corredores, esquinas, etc.).

En los sistemas basados en comportamientos se pueden encontrar principalmente dos arquitecturas:

- Deliberativa. En esta arquitectura, el robot por medio de los sensores toma lecturas del entorno, hace un procesamiento de la información, planea y luego actúa de acuerdo a los resultados obtenidos en la etapa anterior (figura 1.14, a). Este enfoque presenta ciertas desventajas tales como retardos en las respuestas del robot, lo que implica que no existe una relación directa entre los sensores y los efectores del robot, finalmente necesita de un sistema de control complejo (Wicaksono et al., 2009).

- Reactiva. El robot no tiene ningún modelo *a priori* del entorno para tomar alguna decisión, su modelo es el entorno actual y puede producir también comportamientos complejos (Figura 1.14, b) (Bräunl, 2008; Wicaksono et al., 2009)

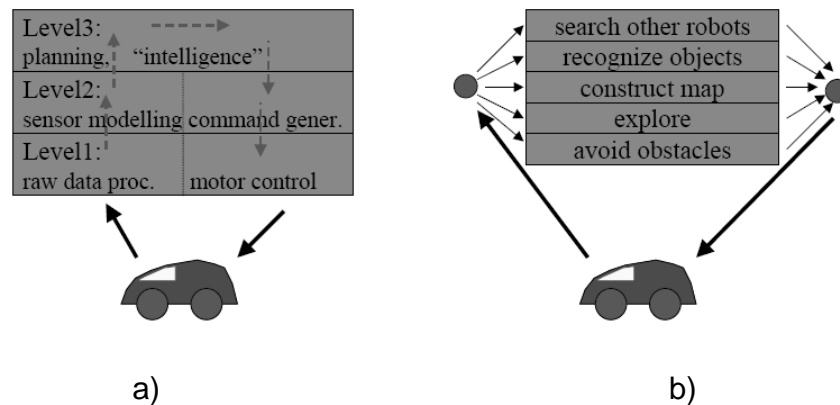


Figura 1.14. Arquitecturas de navegación. a) Deliverativa, b) Reactiva

En la figura 1.14. se pueden presenciar las dos arquitecturas comúnmente utilizados en robótica móvil.

La arquitectura reactiva mencionada anteriormente, es conocida por la comunidad científica como “Arquitectura basada en comportamientos”, y fue propuesta por Rodney A. Brooks, Member, IEEE. en (Brooks, 1986) donde expone la dinámica de dicha arquitectura la cual está compuesta por capas, donde las capas con más alto nivel pueden subsumir las capas inferiores. La figura 1.15 muestra la arquitectura de subsunción propuesta por Brooks.

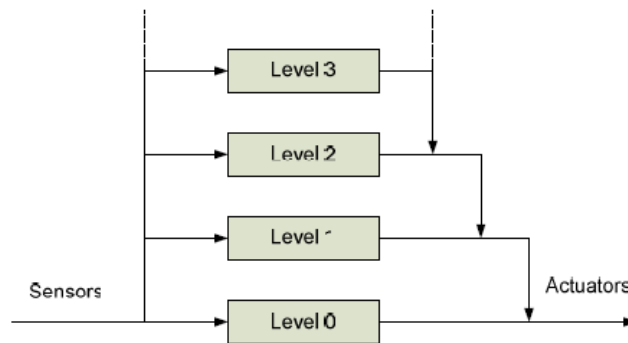


Figura 1.15. Arquitectura de subsunción (Brooks, 1986).

Esta arquitectura propone descomponer tareas complejas de navegación en diferentes subtareas llamadas comportamientos “*Behaviors*” los cuales toman información del entorno de forma paralela y se encuentran conectados directamente a los sensores y actuadores del sistema. El proceder que proporciona la arquitectura mencionada hace innecesario el uso de controladores sofisticados los cuales pueden ser reemplazados por un microcontrolador de bajo costo (Bräunl, 2008; Dongshu, Yusheng, & Wenjie, 2011; Hagra et al, 1999; Qing-yong et al., 2009; Wicaksono et al., 2009). La figura 1.16 muestra el trabajo en paralelo de las tareas.

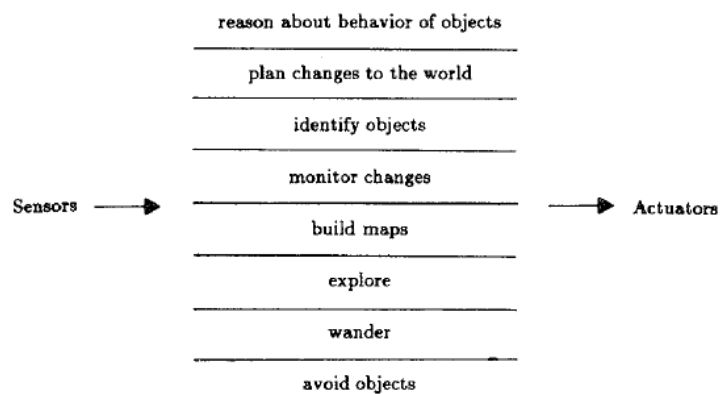


Figura 1.16. Tareas en paralelo (Brooks, 1986).

El paralelismo en la arquitectura presentada por Brooks conlleva a un problema el cual está asociado a “cual comportamiento debería tener el control de los efectores en un instante de tiempo”. La razón principal de este problema es que al estar los

comportamientos conectados directamente a los sensores y actuadores, cada uno puede tomar lecturas en forma paralela del entorno, realizar la acción de control al mismo tiempo y tratar de ejecutarla en paralelo en búsqueda de controlar los efectores al mismo instante, lo cual genera conflictos entre comportamientos, este problema es conocido como la acción de selección (Bräunl, 2008; Dongshu et al., 2011).

Para solucionar estos conflictos entre comportamientos se hace necesario entonces contar con un árbitro (arbitraje) o métodos de fusión los cuales son mecanismos de coordinación de comportamiento que pueden resolver dicho problema (Paolo Pirjanian, 1999; Wicaksono et al., 2009). En las figuras 1.17 se puede ver clases de mecanismos de coordinación, y en la figura 1.18 muestra los mecanismos de coordinación de comportamientos.

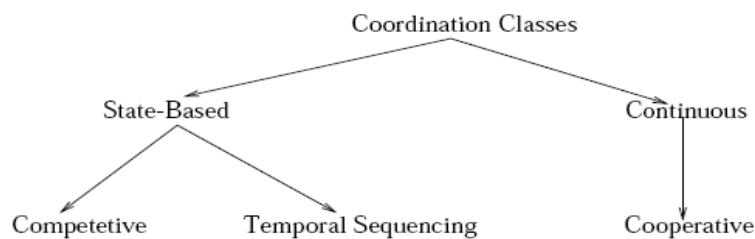


Figura 1.17. Clases de mecanismos de coordinación (MacKenzie et al., 1997).

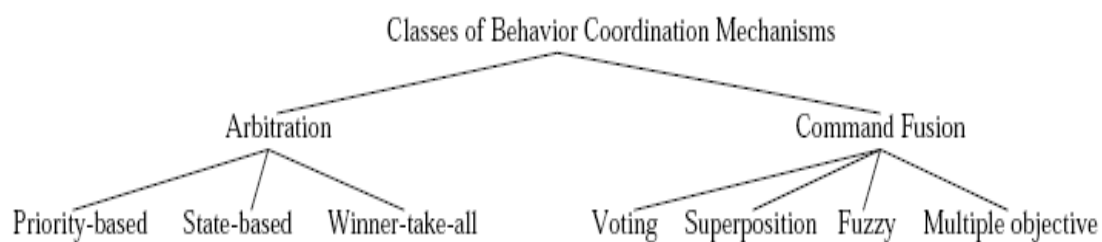


Figura 1.18. Mecanismos de coordinación de comportamiento (Pirjanian, 1999).

Entre los mecanismos de coordinación de comportamientos que utilizan arbitración como coordinación, se puede encontrar como ejemplo la arquitectura presentada por *Brooks* como acción de selección debido a la dinámica que presenta. Ésta se clasifica como un método competitivo basado en prioridades en donde los

comportamientos compiten por controlar los motores del robot y el comportamiento con mayor prioridad, tomará el control de los efectores del robot móvil (Wicaksono et al., 2009).

Si bien el método de coordinación de comportamientos competitivo funciona bien para algunos casos, su naturaleza lo conlleva a describir movimientos bruscos lo que significa fuertes transiciones entre comportamientos y por ende respuestas inadecuadas del robot móvil que pueden resultar en navegaciones no deseadas.

En contribución a la solución del problema de selección se ha planteado otra alternativa. Este método es considerado cooperativo debido a la capacidad que tiene de fusionar las salidas de los comportamientos por diferentes métodos (ver figura 1.18, Command fusion). Esto significa que más de un comportamiento pueden estar activos y contribuir juntos a la acción de control en pro de una navegación exitosa.

Entre los métodos cooperativos se destaca el uso de Campos potenciales como método de superposición propuesto por Arkin en el cual más de un comportamiento puede efectuar acciones de control, lo cual resulta en mejor respuesta por parte del robot, aumentando la precisión en sus movimientos pero aumenta la complejidad frente al método competitivo de Brooks (Abreu & Correia, 1999; R.C. Arkin, 1987; Wicaksono et al., 2009). En la figura 1.19 se puede observar el esquema para el método de campos potenciales.

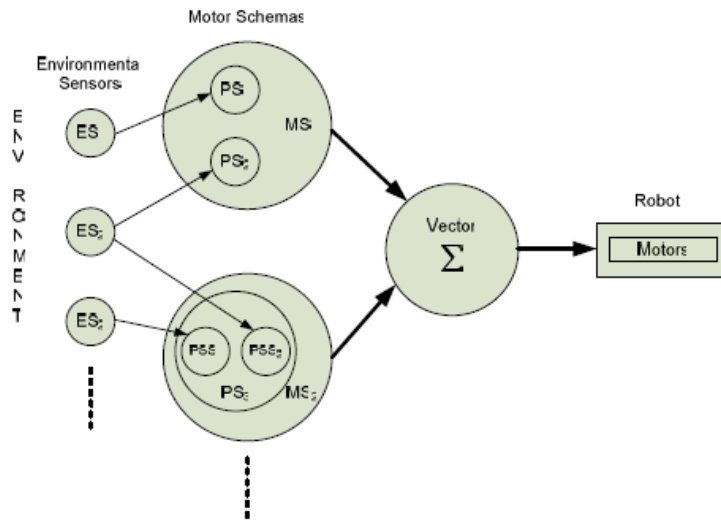


Figura 1.19. Esquema para el método de campos potenciales (Wicaksono et al., 2009)

En los campos potenciales el robot es tomado como un punto en un espacio el cual se moverá debido a las fuerzas de atracción o repulsión de los obstáculos presentes el ambiente de navegación. La acción de selección en este método resulta como el movimiento en la dirección que indica un gradiente en el espacio. El resultado del uso de campos potenciales produce respuestas suaves y mejor precisión del robot, lo que conlleva a movimientos lentos, gran demanda de tiempo en búsqueda del objetivo (Wicaksono et al., 2009).

Otro método cooperativo de fusión ampliamente utilizado es la lógica difusa la cual consiste en la fusificación de las variables de entrada (sensores) a valores de pertenencia que posteriormente serán fusionados con otros valores de pertenencia de otras variables. La combinación de las variables fusificadas se hace mediante set de reglas las cuales se constituyen como un comportamiento que operará con los valores y tomaran una decisión que posteriormente será defusificada como un valor real que será interpretado por los actuadores como la acción de control.

En la figura 1.20 se aprecias un esquema de coordinación de comportamientos basado en lógica difusa (Adriansyah & H. M. Amin, 2005; Wicaksono et al., 2009).

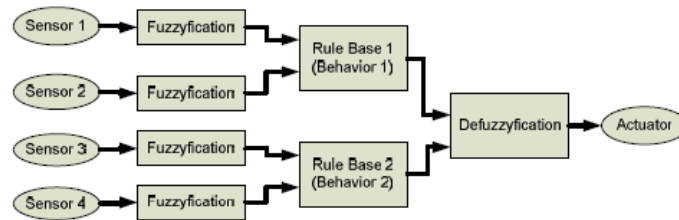


Figura 1.20. Esquema de coordinación de comportamientos basado en lógica difusa (Wicaksono et al., 2009).

Diferente a la lógica difusa, podemos encontrar métodos cooperativos emergentes como la coordinaciones de comportamientos mediante redes neuro-difusas, las cuales atacan el problema de selección asignando pesos a los comportamientos que están activos en un instante de tiempo, para posteriormente ser combinados mediante los principios que aplican las redes neuronales y tomar la decisión más adecuada (Song & Lin, 2006). La figura 1.21 presenta el esquema de coordinación utilizado con redes neuro-difusas.

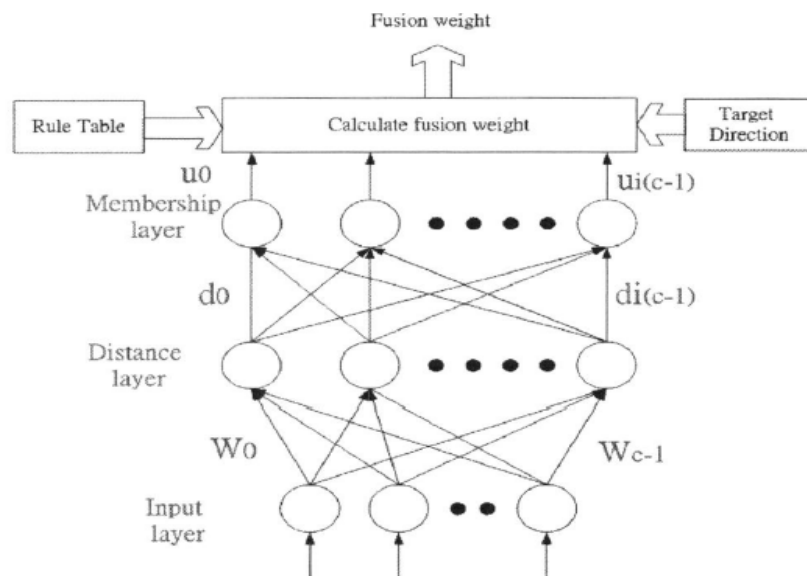


Figura 1.21. Esquema de coordinación de comportamientos neuro-difuso (Song & Lin, 2006)

Otros métodos también conocidos son:

- Mecanismos de Votación (Rosenblatt, 1995).
- Mezcla dependiente de contexto (Safiotti, 1997).
- Coordinación de comportamientos multi-objetivo (P. Pirjanian & Mataric, 2000), entre otros. En (Pirjanian, 1999) se pueden encontrar diferentes aportes en pro de la solución del problema de coordinaciones de comportamientos.

Teniendo en cuenta los antecedentes presentados anteriormente sobre las arquitecturas deliberativas y reactivas (competitivas y cooperativas), podemos también encontrar arquitecturas híbridas de control que combinan ambos panoramas. El objetivo de estas estructuras es buscar un equilibrio entre las dos partes para proporcionar al robot la capacidad de planear y reaccionar rápidamente ante el entorno que navega (Liu & Kubota, 2007; Tang et al., 2006; Zhang et al., 2006). Aunque este tipo de arquitecturas son robustas y ofrecen resultados muy satisfactorios en ambientes tanto estáticos como dinámicos, se debe tener en cuenta que el costo computacional del dispositivo de procesamiento es más alto. Las figuras 1.22 y 1.23 muestran por medio de diagramas de flujo como se componen las diferentes estructuras.

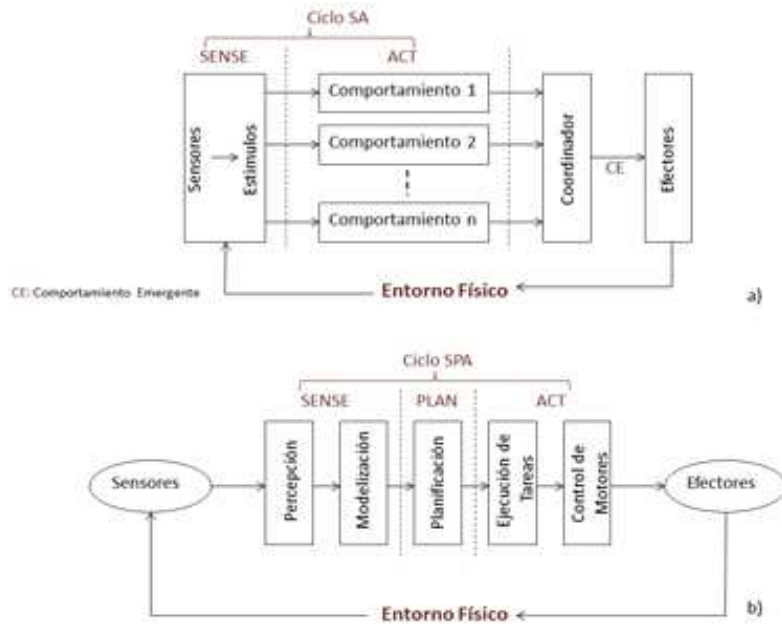


Figura 1.22. a) Arquitectura reactiva b) arquitectura deliberativa (Acosta Amaya, 2010).

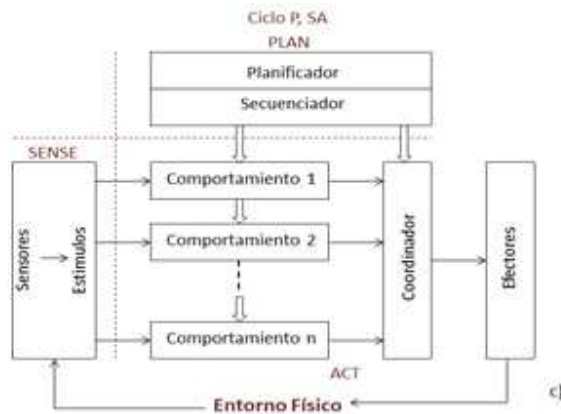


Figura 1.23. Arquitectura híbrida (deliberativa-reactiva) (Acosta Amaya, 2010).

1.9. METODOLOGÍA PROPUESTA

El derrotero planteado para el desarrollo del esquema de coordinaciones de comportamiento para la navegación del robot móvil inicia con la primera etapa, caracterización de los comportamientos más representativos implementados en los

prototipos robóticos para la navegación y exploración de entornos interiores. Ya determinados los comportamientos se procede en una segunda etapa que consta de la construcción de un conjunto que garantice una navegación segura al autómata móvil en el entorno al cual es expuesto. En la tercera etapa del proyecto se diseña un esquema de coordinación el cual integra el conjunto básico de comportamientos establecidos con anterioridad y como cuarta y última etapa del proyecto, se evalúa el desempeño de la arquitectura de control de la plataforma robótica integrando el esquema de coordinación de comportamientos desarrollado y analizando las trayectorias descritas por el robot móvil en su navegación y exploración.

La metodología anteriormente mencionada está constituida por cuatro etapas, las cuales se descomponen en una serie de actividades para cada una, a continuación se describen las actividades relacionadas a las etapas de la metodología:

Primera Etapa – Caracterización los comportamientos más representativos.

Actividad 1. Selección de los comportamientos más utilizados para robots móviles de pequeñas dimensiones.

Actividad 2. Determinan las características de cada comportamiento.

Actividad 3. Construcción de los comportamientos.

Segunda Etapa – Determinar un conjunto de comportamientos necesario para una navegación segura del autómata móvil.

Actividad 1. Prueba de los comportamientos independientemente.

Actividad 2. Evaluación del desempeño de los comportamientos.

Actividad 3. Clasificaciones según el desempeño del comportamiento.

Actividad 4. Construcción de un conjunto que asegure una navegación fiable al robot móvil.

Tercera Etapa – Diseño de un esquema de coordinación de comportamientos.

Actividad 1. Diseño de un esquema de coordinación.

Actividad 2. Integración de conjunto de comportamiento con el esquema de coordinación.

Actividad 3. Pruebas del esquema de coordinación.

Cuarta Etapa – Evaluación del desempeño de la arquitectura de control.

Actividad 1. Evaluación del esquema de coordinación de comportamiento en la plataforma robótica móvil.

Actividad 2. Asegurar una navegación fiable al robot evitando cualquier tipo de colisión.

Actividad 3. Toma de datos de la navegación y explotación del robot móvil.

Actividad 4. Evaluar el desempeño del sistema observando las trayectorias descritas por el robot en la navegación de un entorno cerrado y estático.

1.10. CONCLUSIONES DEL CAPÍTULO

Si bien los diferentes tipos de arquitecturas deliberativas y reactivas descritas en el apartado anterior el investigador de este trabajo infiere en que es más conveniente utilizar una arquitectura reactiva basada en comportamientos utilizando un mecanismos de fusión difuso debido a las siguientes razones:

1. La lógica difusa tiene la capacidad de manejar los problemas de forma intuitiva maniobrando en espacios donde la adquisición de datos por parte de los sensores del robot son difícil de obtener o en casos extremos es incierta.
2. La información de los sensores se puede trabajar fácilmente usando variables lingüísticas que ayudan a una mejor interpretación del problema.
3. Una arquitectura difusa basada en comportamiento puede realizar inferencias online de una manera eficiente utilizando los datos adquiridos en circunstancias inciertas de movimientos y sensado.

4. La capacidad de análisis que brinda el uso de la lógica difusa puede llevar al diseño de controladores robustos en circunstancias de incertidumbre, acogiendo procedimientos del control convencional como el caso de los controladores difusos P, PI, PD y PID.
5. La interpretación de una variable física descompuesta, la cual es resultado del conjunto de variables lingüísticas ayuda a un mejor entendimiento y por lo tanto conlleva a una codificación más sencilla en términos de sentencias IF-THEN en dispositivos de medianas prestaciones como son los microcontroladores.
6. Una arquitectura difusa basada en comportamiento puede propicia acciones de control y transiciones suaves entre activación de comportamientos en la arquitectura diseñada.

Capítulo 2. Descripción de la Plataforma Robótica

En el siguiente apartado se presenta una descripción global del prototipo robótico utilizado para realizar las pruebas de navegación. El prototipo robótico presentado en la figura 2.1 cuenta con cuatro módulos los cuales están compuestos por electrónica y componentes necesarios para realizar tareas de navegación en entornos cerrados y estáticos.

Los cuatro módulos serán explicados brevemente a continuación.



Figura 2.1. Plataforma robótica CARLITOS (acrónimo de **C**artographic Intelligent Robot for **SLAM**.)

2.1. MÓDULO DE COMUNICACIONES

En esta etapa de comunicaciones el sistema robótico cuenta con dos dispositivos electrónicos llamado Xbee los cuales tienen la capacidad de comunicarse punto a punto o multipunto con otros dispositivos de su clase y enviar datos por medio de un enlace de frecuencia (RF).

La siguiente figura muestra cómo se efectúa la comunicación entre los dispositivos Xbee en el sistema robótico *CARLITOS*.

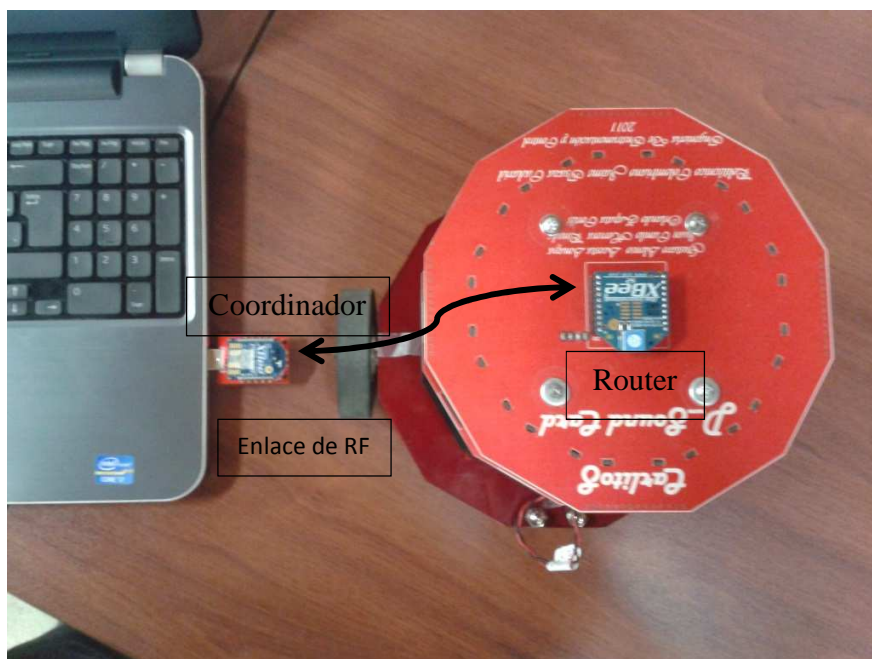
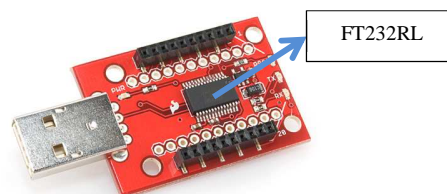


Figura 2.2. Enlace de comunicación Xbee.

Los dos dispositivos Xbee que pertenecen al sistema robótico y que permiten la comunicación Robot-ordenador están configurados como coordinador-router para que exista una comunicación punto a punto. Dicha comunicación esta referenciada por medio de una ID que posee cada dispositivo y es única, la cual sirve como referencia para identificarse entre ellos y así propiciar el canal comunicación.

En adición al sistema de comunicación se agrega una sub-etapa electrónica la cual sirve como medio para que pueda haber un entendimiento entre el Bus Serial Universal (USB) del ordenador y el Xbee, esta sub-etapa electrónica está compuesta por un dispositivo llamado FT232RL que acompañado con una configuración de otros elementos electrónicos tiene la capacidad de convertir los niveles lógicos que maneja el Xbee a los niveles lógicos que maneja el puerto USB. La figura 2.3.a. muestra la sub-etapa electrónica de comunicación Xbee-Ordenador denominada Xbee Explorer Dongle desarrollada por la empresa “Sparfun Electronics” y en La figura 2.3.b muestra el acople de comunicación Xbee-USB mediante el Xbee Explorer Dongle hacia el ordenador.



a.



b.

Figura 2.3. a) Xbee Explorer Dongle; b) comunicación Xbee-USB

El canal que utiliza los dispositivos Xbee para comunicarse internamente con sistema robótico son los canales de transmisión 2 (Tx2) y recepción 2 (Rx2) (la aclaración de los canales es debido a que existe otro canal de transmisión 1 (Tx1) y recepción 1 (Rx1) utilizado por los sensores de ultrasonido). Por estas pistas de

comunicación fluye la información procesada por el microcontrolador ubicado en el módulo de control el cual adquiere los datos suministrados por diferentes dispositivos ubicados en otros módulos (sensores propioceptivos: brújula- encoders, exteroceptivos: sonares) e información general procesada por el microcontrolador o también comandos desde una interfaz serial del ordenador al microcontrolador. En la figura 2.4 se presenta la estructura general de comunicación ordenado-Xbee-Microcontrolador en ambos sentidos.

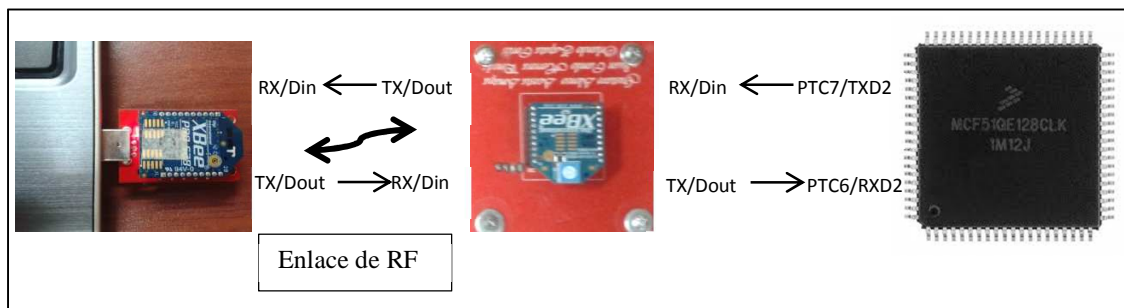


Figura 2.4. Estructura general de comunicación Ordenado-Xbee-Microcontrolador.

Como se puede ver en la figura 2.4 esta estructura de comunicación se denomina semi-dúplex en donde el flujo de información fluye en uno u otro sentido, pero nunca en simultáneo. La razón de esta comunicación es debido a que tanto el ordenador como el microcontrolador pueden transmitir y recibir señales ya sea de *start-stop* o información en general. Esta figura muestra gráficamente como opera la comunicación entre el robot y el ordenador

2.2. MÓDULO DE PERCEPCIÓN

El módulo de percepción o anillo de sonares del robot *Carlitos* está compuesto por un acople de sensores de ultrasonido también llamados sonares los cuales están estratégicamente posicionados en los doce lados de un dodecágono, con el objetivo de disminuir las zonas muertas de detección entre dispositivos. Separadamente, también hacen parte del sistema de percepción una brújula digital ubicada en el

módulo de control y dos *encoders* de cuadratura acoplados a los ejes de motores, estos hacer parte del módulo de motores como dispositivos propioceptivos. Los sonares están interconectados a las pistas de transmisión, recepción, polarización y modo trabajo generales para la tarjeta electrónica. En la figura 2.5 muestra cómo están interconectados los sensores del robot.

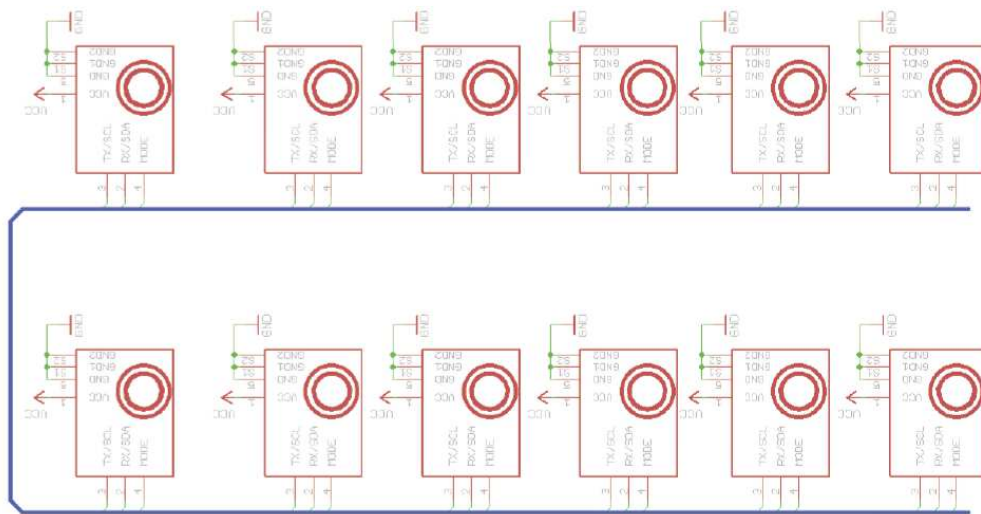


Figura 2.5. Interconexión los sonares.

Se puede observar que los pines de Tx, Rx, mode, Vcc y Gnd están interconectados de manera que todos los sensores estén disponibles para ser utilizados en cualquier momento. La figura 2.6 muestra físicamente el anillo de sonares



Figura 2.6. Anillo de sonares.

Estos sonares están configurados para que trabajen en modo serial, pero también presenta la característica de trabajar en modo I2C el cual es un protocolo de comunicación que trabaja bajo el esquema maestro (microcontrolador)-esclavo (sonares). El diseño de la tarjeta electrónica para este módulo también soporta este modo de trabajo. El cambio de protocolo de comunicación se hace mediante el cambio de posición de dos *jumpers* puestos en la tarjeta electrónica los cuales habilitan esta forma de trabajo. En la figura 2.7 muestra el circuito de cambio de modo de trabajo de UART (serial) a I2C.

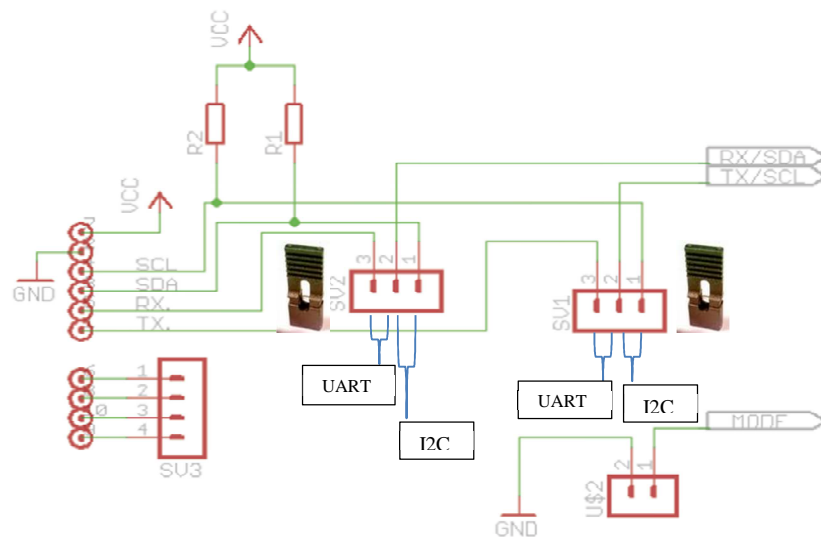


Figura 2.7. Modo de trabajo de UART(Serial) a I2C.

2.2.1. Características del sonar SRF02

La tabla 1 presenta las características de sonar y la figura 2.8 muestra el hardware que constituye el sensor utilizado.

ESPECIFICACIONES TÉCNICAS DEL SONAR SFR02	
Voltaje	5v
Corriente	4mA (típica)
Frecuencia	40KHz

Rango Máximo	600cm
Rango Mínimo	15cm
Ganancia	Control Automático de 64 pasos
Conectividad	Bus I2C, Serial UART
Ajuste (calibración)	Automática en el encendido
Unidades de la medida	µs, cm, pulgadas
Peso	0.16226 oz.
Dimensiones	24mm w, 20 mm d, 17mm h

Tabla 1. Especificaciones técnicas sonar SRF02.

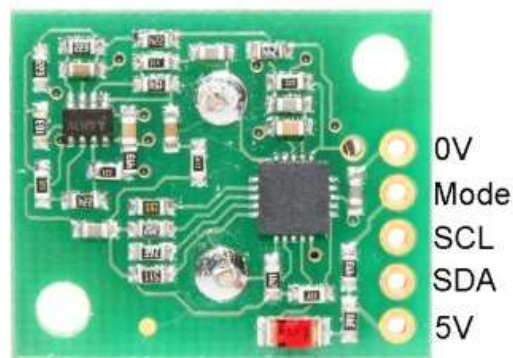


Figura 2.8. Diagrama de pines.

La detección de obstáculos por sonar SFR02 como todo sistema electrónico presenta un rango de funcionamiento. Particularmente, en el reconocimiento de un objeto, el sonar debe encontrar dicho elemento dentro del lóbulo de radiación y no estar a una distancia menor de 16 cm (según especificación la tabla 1), este margen de error está asociada a que el transductor del sensor como Tx y Rx . En la figura 2.9, se muestra el lóbulo de radiación del dispositivo.

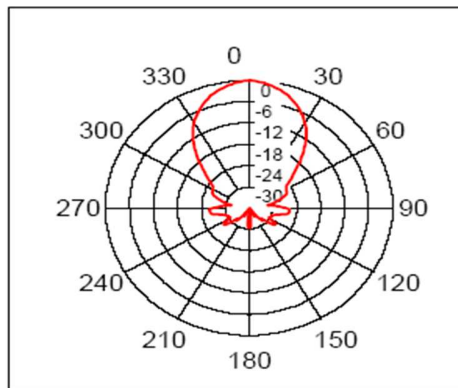


Figura 2.9. Lóbulo de radiación.

2.2.2. Fenómenos asociados.

Las lecturas se pueden ver afectadas por los siguientes fenómenos dando como resultado lecturas espurias por parte de los dispositivos (KARSULOVIC et al., 2008). Los fenómenos son los siguientes:

- **Scattering.** Proceso por el que la onda acústica es dispersada en multitud de direcciones tras chocar con un objeto.
- **Reflexiones especulares.** Este fenómeno se presenta cuando la onda acústica se refleja en la superficie objeto y esta llega con una inclinación superior a los 30 grados generando lecturas erróneas, en algunos casos la lectura nunca retorna debido a fenómenos múltiples de reflexión.
- **Absorción.** La absorción del ultrasonido ocurre cuando la energía de las vibraciones de la onda es disipada en su movimiento molecular interno y éste último en calor.

2.3. MÓDULO DE CONTROL

El módulo de control cuenta con canales de comunicación para los diferentes dispositivos mencionados anteriormente, *led* de señalización y accionamientos de

jumper para enciendo del sistema, electrónica de regulación de voltajes para poder realizar el acople ente los diferentes elementos utilizados. También cuenta con un *buzzer* para que alerte a las personas con un sonido agudo que el robot se encuentra en marcha, canales de programación y el microcontrolador es que quizás el dispositivo más importante de todo el sistema, puesto que en él recae los diferentes algoritmos de navegación, manejo de periféricos para sensores, motores, comunicaciones, etc.

El microcontrolador como dispositivo central del robot tiene la capacidad de realizar tareas complejas como pre-procesamientos y procesamiento, inferencia para tomar decisiones, acciones de control entre otras, las cuales conllevan un gran esfuerzo computacional. En la figura 2.10 se presenta el módulo de control donde de hace una descripción grafica del mismo.

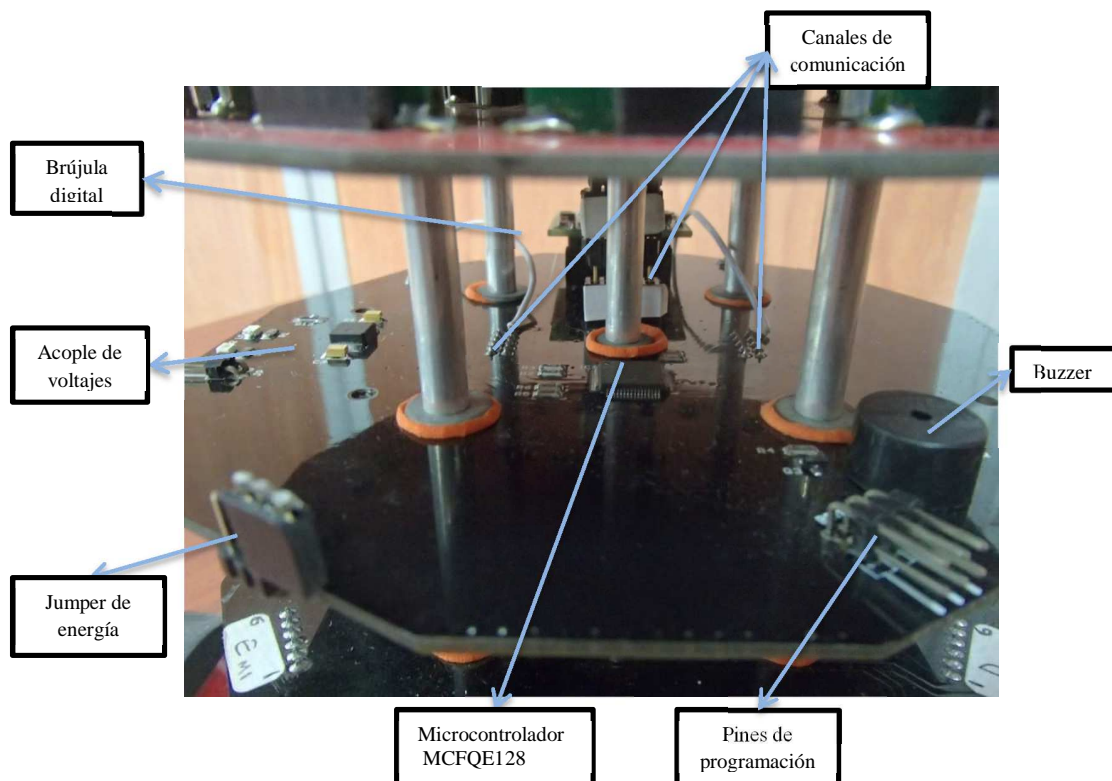


Figura 2.10. Módulo de control.

Aunque en este módulo se cuenta físicamente con la brújula digital ubicada en la parte frontal del robot, esta hace parte del sistema de percepción. El microcontrolador como eje central del robot está asociado entonces a todos los elementos que constituyen el robot ya sea por los canales de Tx y Rx o los periféricos propios conectados a los diferentes dispositivos electrónicos. En la figura 2.11 se presenta el diagrama esquemático de conexiones del microcontrolador hacia los demás elementos del sistema robótico.

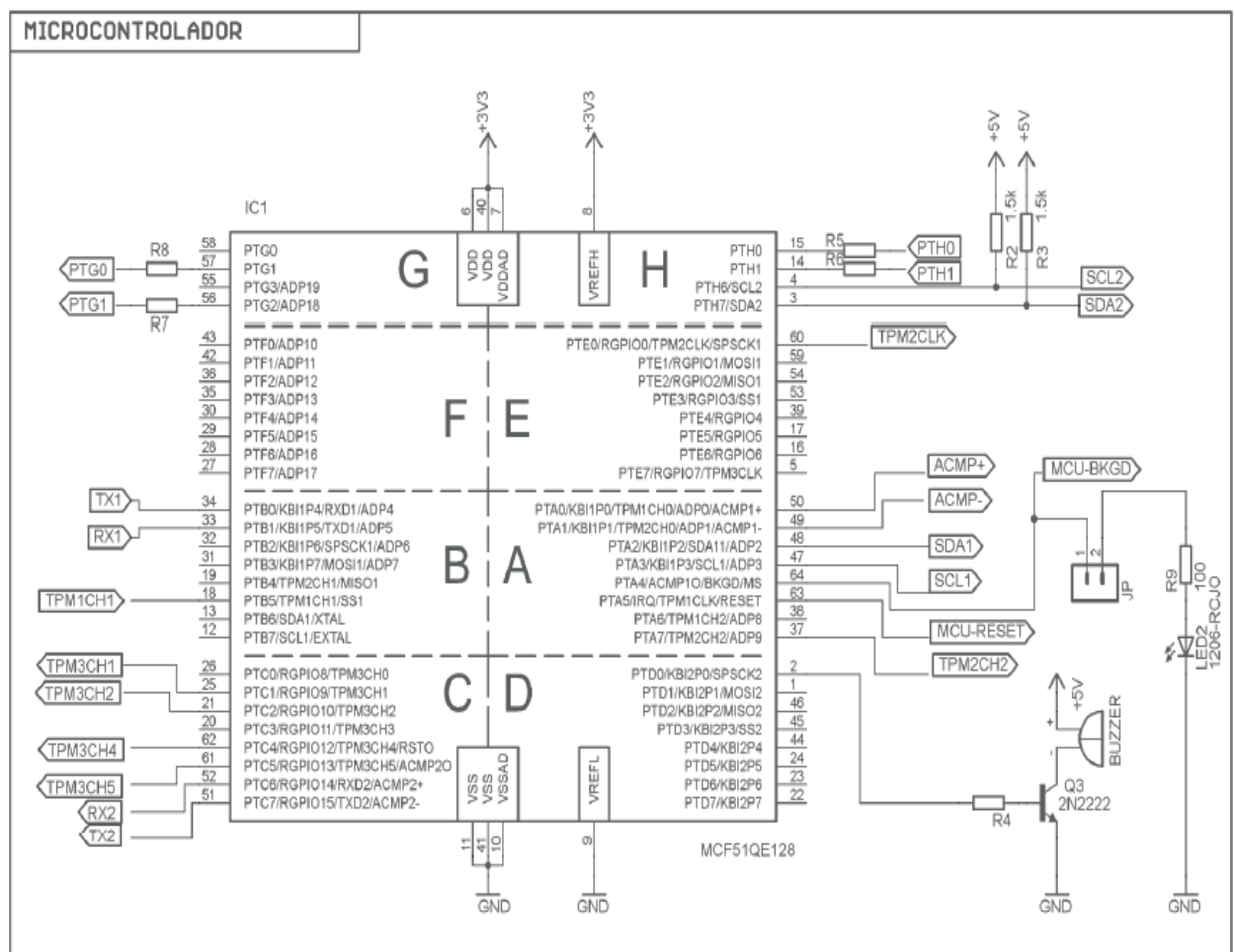


Figura 2.11. Conexiones del microcontrolador.

Se precia entonces, las múltiples etiquetas sujetas a los pines del microcontrolador especificando los canales de los periféricos de comunicaciones, comparador análogo, salidas digitales, programación, polarización entre otras.

La polarización de este microcontrolador es suministrada por dos pares de pilas AAA que con acople de voltajes se provee 3.3 voltios necesarios para el microcontrolador el cual pertenece a la familia COLDFIRE V1 de la empresa *FREESCALE™*. En la figura 2.12 se muestran las especificaciones globales del microcontrolador.

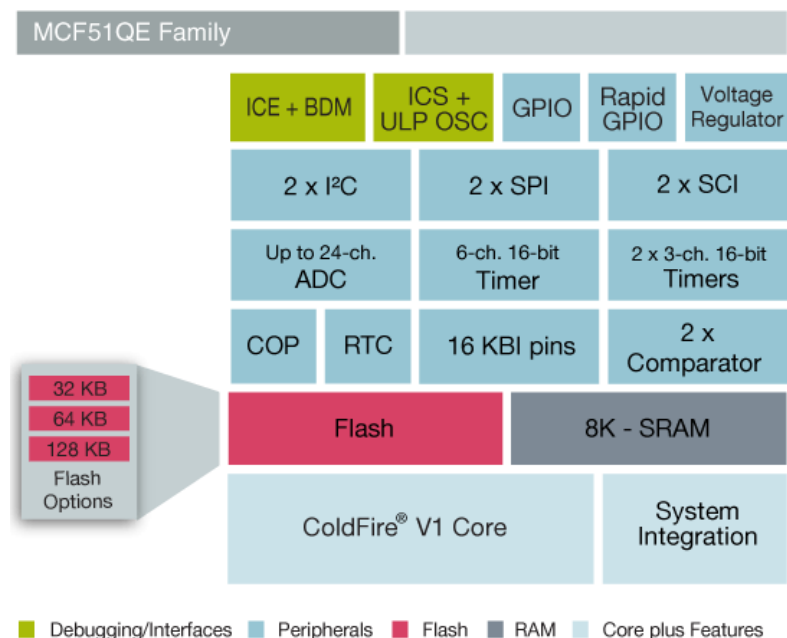


Figura 2.12. Especificaciones globales del microcontrolador (Freescale, 2013).

2.4. MÓDULO DE ACTUADORES

Para concluir con la descripción del sistema robótico utilizado, se tiene por último el módulo de actuadores el cual está constituido a groso modo por dos *drivers* integrados en un solo *chip* para el manejo de motores con su electrónica externa

necesaria para su operación, dos motoredutores, dos baterías de 7.4 voltios en serie para un total de 14.8 voltios que energizan dichos motores y que en simultaneo proporcionan la alimentación de los sensores de ultrasonido y otros dispositivos con su debido acople de tensión. La figura 2.13 muestra el módulo de actuadores donde se hace una descripción de esta.

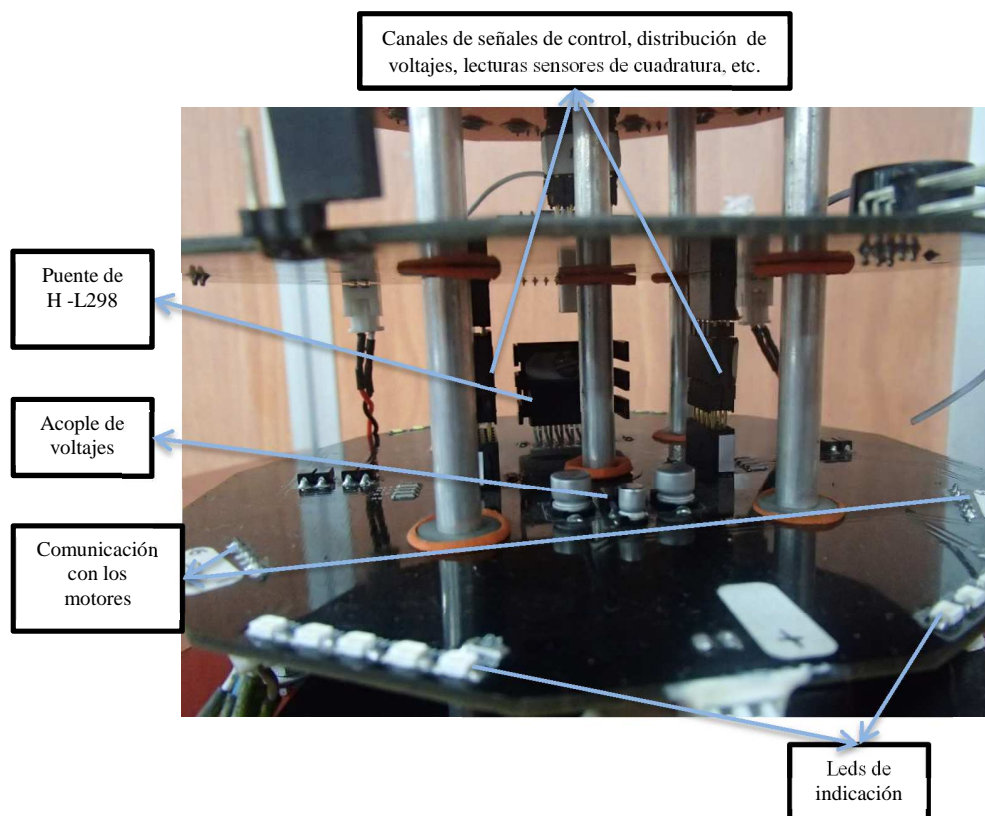


Figura 2.13. Módulo de actuadores

El integrado L298 cumple un papel fundamental en esta tarjeta, puesto que es el principal encargado del manejo de los motores. Dependiendo de los estados lógicos de sus pines de habilitación y la manipulación de los pines de entrada para cualquiera de los dos *drivers* de del integrado, se puede controlar el sentido de giro

del motor que se desea. En la figura 2.14 se muestra como está configurado el puente en H L298 para el robot *CARLITOS*.

Este tipo de configuración permite al programador en cualquier momento poder reconfigurar los pines del microcontrolador que están conectados a las entrada de los dos *drivers* y cambiar bien sea el sentido de giro de los motores o alterar la velocidad por medio de un PWM (Pulse Width Modulation) *On-line* u *Off-line*.

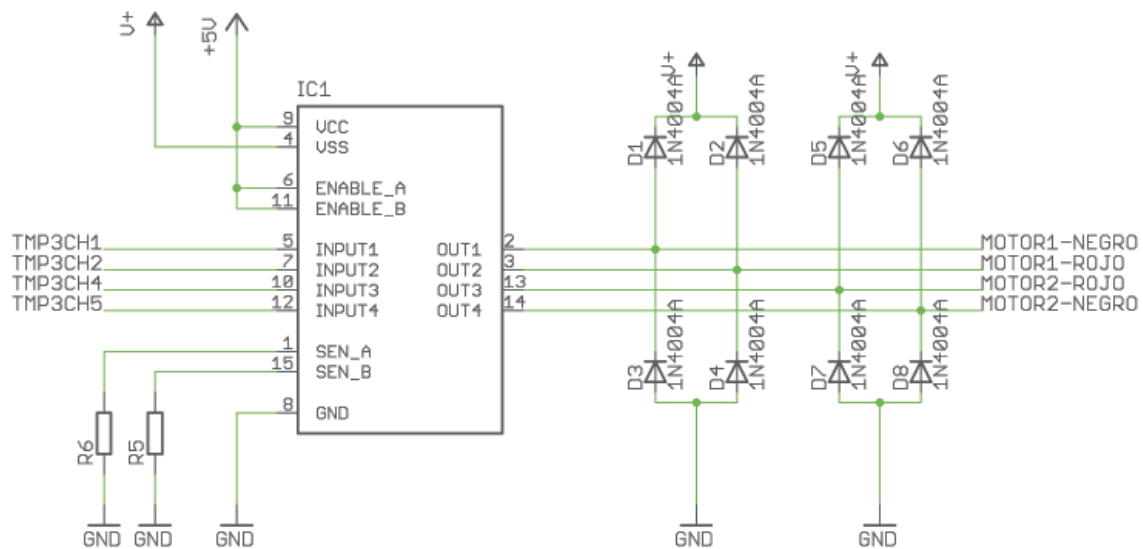


Figura 2.14. Esquemático configuración puente en H - L298

En el modo de operación del puente es:

- Estando siempre habilitados los *drivers* (enable_A y enable_B; puestos a 5 voltios) se indica que siempre van a estar en funcionamiento los motores, se procede entonces a manipular las entradas para controlar el sentido de giro o velocidad de cada motor.
- Los pines asociados a TPM3CH1/PTC1-Motor izquierdo y TPM3CH4/PTC4-Motor derecho, sirven como pines de dirección que definirán el sentido de giro de los motores dependiendo del estado lógico en el que se encuentren (0 o 5 voltios).

- Por los canales TPM3CH2/PTC2-Motor izquierdo y TPM3CH5/PTC5-Motor derecho, se induce un PWM a una frecuencia de 1KHz, que según las variaciones del ancho de pulso (Duty Cycle) aumentará o disminuirá la velocidad en los motores.

En la figura 2.15 se muestra una interpretación grafica de los pasos mencionados anteriormente.

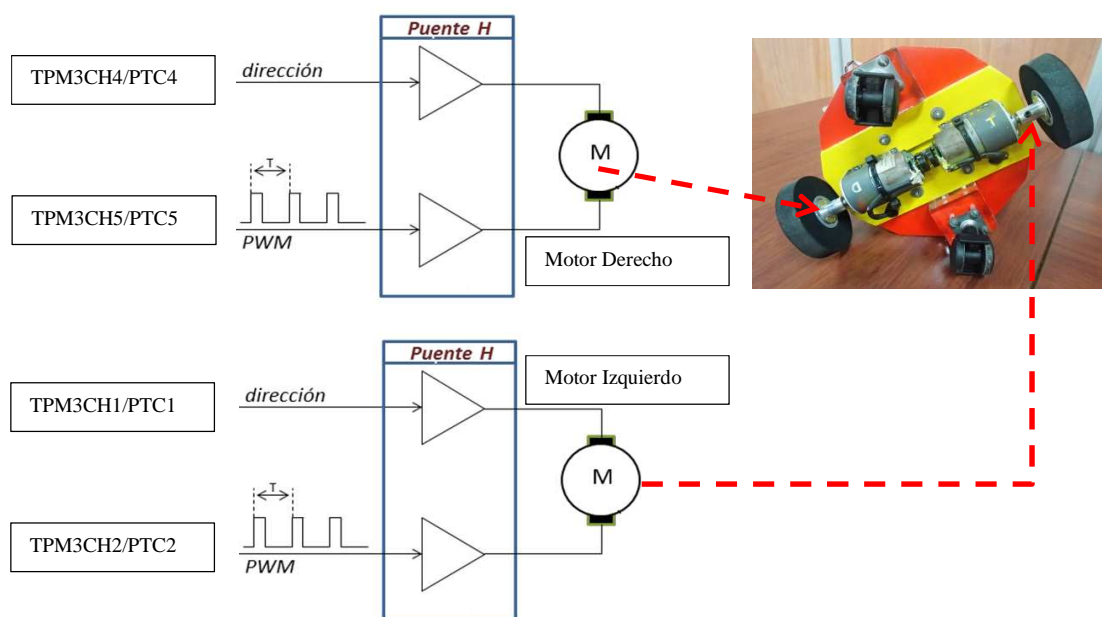


Figura 2.15. Accionamiento de motores.

Como puede verse en la figura 2.15, la plataforma robótica cuenta con dos *casterballs* como soporte y dos motores en configuración diferencial. La dinámica de estos motores en tal configuración se puede modelar de la siguiente manera.

En la figura 2.16, se muestra el desplazamiento relativo entre instantes de muestreo, en el cual se presentan las relaciones del arco recorrido por el centro del robot, orientación del robot y distancia entre los puntos de apoyo de las ruedas motrices del robot.

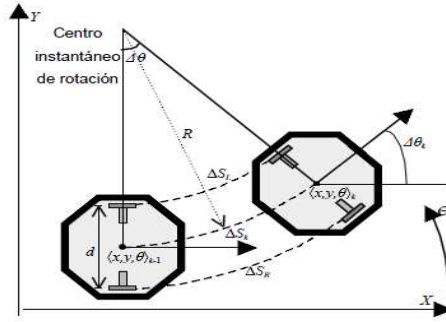


Figura 2.16. Cinemática en robots diferenciales(Navarro et al., 2007).

De la figura 2.16 se obtiene las siguientes relaciones,

$$\Delta S_k = (\Delta S_R + \Delta S_L) / 2 \quad (1)$$

$$\Delta \theta_k = (\Delta S_R - \Delta S_L) / d \quad (2)$$

Donde ΔS_R y ΔS_L de las ecuaciones (1) y (2) respectivamente son los recorridos de la rueda izquierda y la rueda derecha; ΔS_k y $\Delta \theta_k$ son el arco recorrido por el centro del robot y el giro diferencial respectivamente y d es la distancia entre los puntos de apoyo de las ruedas motrices. Entonces ajustando el RTC (Real Time Counter) módulo de reloj de tiempo real y estableciendo el tiempo de muestreo constante de un segundo se puede estimar la posición y la orientación de *CARLITOS* mediante las siguientes ecuaciones en diferencia (Danilo Navarro et al., 2007).

$$\begin{cases} x_k = x_{k-1} + \Delta S_k \cos(\theta_{k-1} + \Delta \theta_k / 2) \\ y_k = y_{k-1} + \Delta S_k \sin(\theta_{k-1} + \Delta \theta_k / 2) \\ \theta_k = \theta_{k-1} + \Delta \theta_k \end{cases} \quad (3)$$

Donde x_k , y_k son la posición del robot y θ_k es la orientación.

2.5. CONCLUSIONES DEL CAPÍTULO

- Las características que presenta el robot descrito en este capítulo hacen que el autómata sea apto para realizar tareas de navegación en entornos

interiores de trabajo, ya que se cuenta con los elementos necesarios para desarrollar estas labores.

- Las experiencias de trabajo con el microcontrolador MCFQE128 garantiza que el dispositivo como herramienta es capaz de soportar algoritmos complejos, realizar tareas rápidamente asociadas a eventos externos o internos como interrupciones por recepción de datos, transmisión de información vía serial, interrupciones de tiempo real entre otras.

Capítulo 3.

Objetivo 1: Comportamientos en Robots Móviles

Como se planteó inicialmente en este trabajo de investigación uno de los problemas asociados al control basado en comportamientos es el diseños de ellos mismos, estos problemas están relacionados con cómo debe reaccionar el robot en su entorno y que decisiones debe tomar (Adriansyah & H. M. Amin, 2005).

En esta sección se aborda la descripción global de los comportamientos habitualmente utilizados en la navegación con robots móviles. Esta descripción es soportada mediante la búsqueda de literatura referente al tema, para luego realizar un inventario de comportamientos. De este inventario se seleccionaron los posibles candidatos los cuales formaron el conjunto base de comportamientos que fueron programados en el sistema robótico descrito en el capítulo 2.

En la revisión realizada se encontró que comúnmente se hace uso de tres o cuatro comportamientos base en cada robot. A continuación serán descritos el inventario de comportamientos consultados.

3.1. INVENTARIO DE COMPORTAMIENTOS

3.1.1. Comportamiento 1: Seguimiento de Paredes (Wall Following Behavior)

El comportamiento de seguimiento de pared puede ser tratado de dos maneras: usualmente éste se descompone como seguimiento de pared derecha o seguimiento de pared izquierda por separado, o también puede ser trabajos en simultáneo.

Cuando este comportamiento realiza la detección de paredes por separado, se habla de dos comportamientos los cuales son diseñados a partir de los sensores que dispone en cada lado, usualmente uno o dos sensores y dependerá de las lecturas de estos dispositivos. La figura 3.1 muestra esquema de seguimiento de paredes por separado.

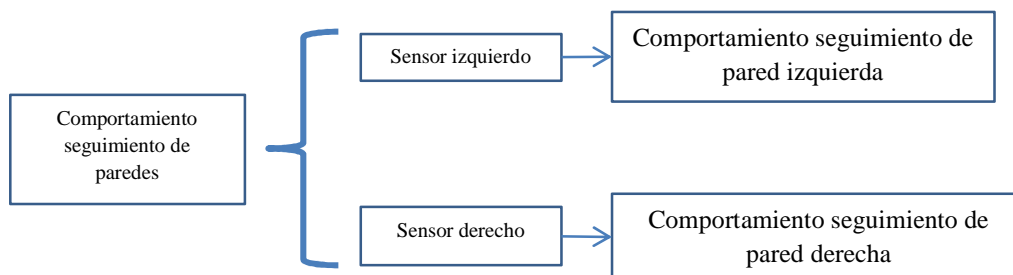


Figura 3.1. Esquema de seguimiento de paredes por separado

En este esquema presentado, cada comportamiento utiliza como variable de entrada la lectura de los sensores correspondientes a cada lado y toma sus decisiones por separado para afectar las revoluciones del motor que controlan. En contraste a este esquema se encuentra detección de paredes en simultáneo, donde el robot toma las lecturas de los sensores correspondientes de cada lado como entradas al mismo controlador para hacer inferencia y de acuerdo a esta inferencia se toma acción en un motor o ambos. La figura 3.2 muestra el esquema utilizado en la detección de paredes en simultáneo y la figura 3.3 ilustra la dinámica del comportamiento en la detección en simultáneo.

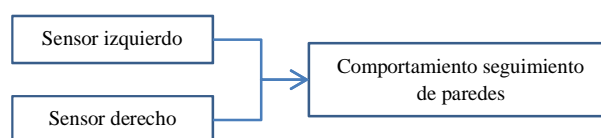


Figura 3.2. Detección de paredes en simultáneo.

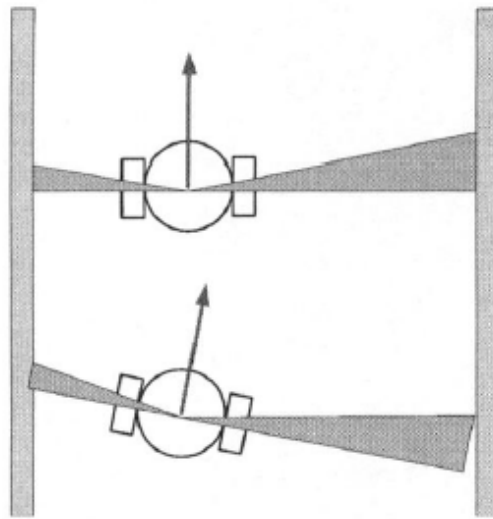


Figura 3.3. Detección en simultáneo (Song & Lin, 2006).

Los sensores comúnmente utilizados para el diseño de este comportamiento son sensores de tipo infrarrojo o ultrasonido, los cuales son sensores de distancias aptos para esta tarea de cálculo de distancia a objetos, en (Dongshu et al., 2011; Farooq et al., 2010; Qing-yong et al., 2009; Song & Lin, 2006) se puede consultar esquemas de diseño para la detección de paredes con sensores de ultrasonido .

3.1.2. Comportamiento 2: Evasión de Obstáculos (Obstacle Avoidance Behavior)

El diseño del comportamiento de evasión de obstáculos esta soportado esencialmente por los sensores frontales que posee el robot móvil y en esencia lo que busca es que el robot no sufra ninguna colisión frontal o a sus alrededores contra un objeto cualquiera. Esta evasión de obstáculos al igual que el seguimiento de paredes se soporta por medio de sensores de distancia tales como los sonares e infrarrojos. En la figura 3.4 se puede ver un ejemplo de un esquema de evasión de obstáculos frontales utilizando sonares y sensores infrarrojos.

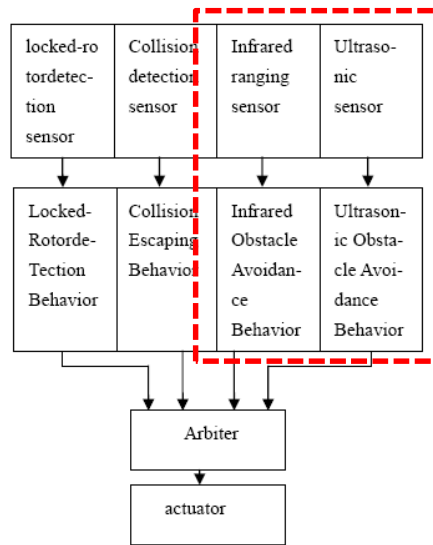


Figura 3.4. Ejemplo de esquema de evasión de obstáculos frontales (Peng & Yumei, 2009).

En la figura 3.4 se ve entonces que el diseño de este comportamiento se realizó a partir del uso de dos sensores de distancia de diferente tipo. Se infiere entonces que la complejidad de este comportamiento también dependerá de cuantos sensores frontales sean utilizados para detectar los objetos. La acción de control dependerá del ambiente y de las preferencias de programador. Esto quiere decir que dependiendo de los objetos que presente el entorno de prueba y las distancias adquiridas por los sensores, el programa decidirá si el robot deberá girar a la izquierda o derecha cambiando las revoluciones de los motores para bordear el obstáculo que se interpone en la navegación del robot. En la figura 3.5 se presenta la simulación de un robot evadiendo un obstáculo.

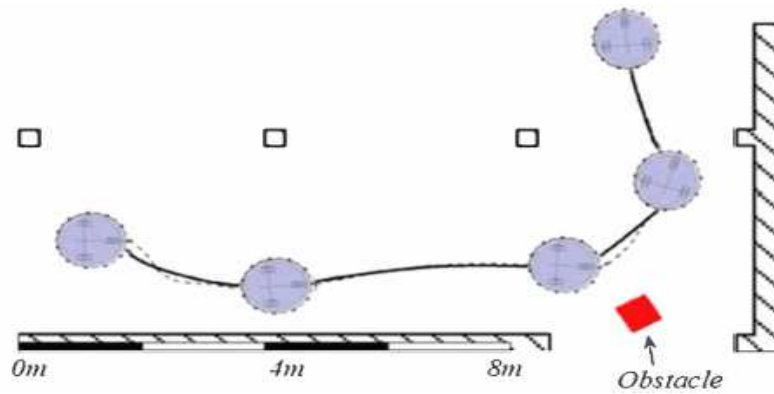


Figura 3.5. Simulación de un robot evadiendo un obstáculo(Shim et al., 2011)

3.1.3. Comportamiento 3: Búsqueda de Objetivos (Goal Seeking Behavior)

En este comportamiento el robot tiene como finalidad buscar un objetivo o meta. En la búsqueda de este objetivo el robot se apoya en el diferencial de ángulo que existe entre la posición actual del robot y la posición del objetivo. En este comportamiento, la entrada procesada por el algoritmo es el diferencial de ángulo existente y el cual debe ser corregido a lo largo de la navegación para cumplir su tarea. En la figura 3.6 muestra la representación de este comportamiento.

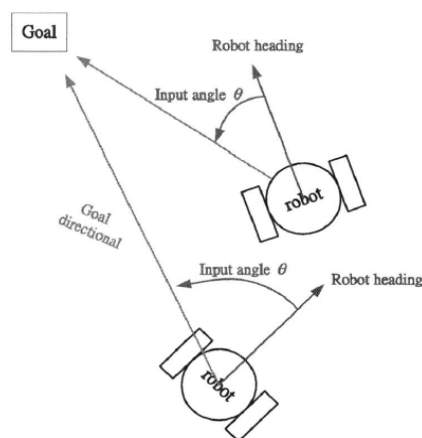


Figura 3.6. Goal Seeking Behavior (Song & Lin, 2006).

3.1.4. Comportamiento 4: Evasión de Estancamiento (Deadlock Disarming Behavior)

Cuando el robot en su navegación entra en lugares estrechos donde no hay salida como pasillos estrechos u objetos en forma de U o V se le dificulta su navegación. Este comportamiento evita que el robot caiga en estos estancamientos evadiendo estos ambientes geométricos usando sus sensores frontales y laterales (Qing-yong et al., 2009). La inferencia que hace el robot es que tan pronto detecta la presencia de un entorno con las especificaciones anteriormente presentadas, él deberá salir lo más pronto posible realizando grandes giros de orientación, lo que implica un cambio en las revoluciones del motor a ser controlado. La figura 3.7 muestra el comportamiento de evasión de estancamiento.

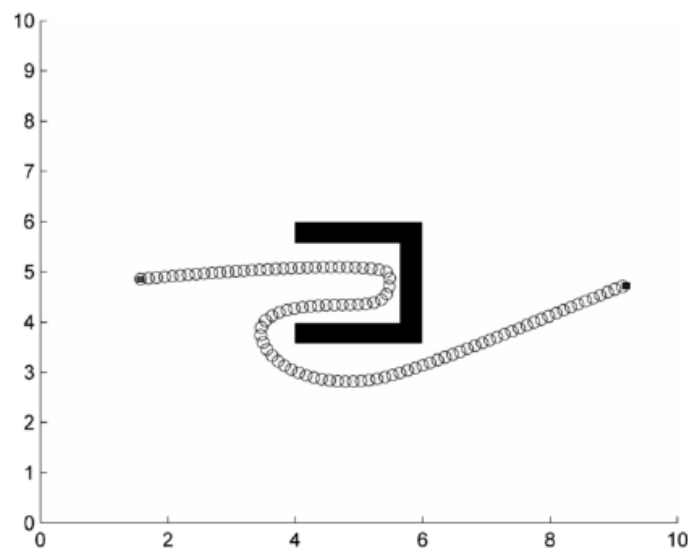


Figura 3.7. Deadlock disarming behavior (Dongshu et al., 2011).

3.1.5. Comportamiento 5: Comportamiento de Emergencia (Emergency Behavior)

El comportamiento de emergencia es la conducta con mayor prioridad que se encuentra en los esquemas reactivos de control basados en comportamientos. En la figura 3.8 se muestra que si un objeto está excesivamente cerca del robot o el robot por alguna razón pierde el control y sus sensores detectan un objeto muy cercano traspasando un umbral de distancia límite, el robot debe parar inmediatamente para no sufrir ningún daño y no enfrentarse al objeto (Thongchai et al., 2000).

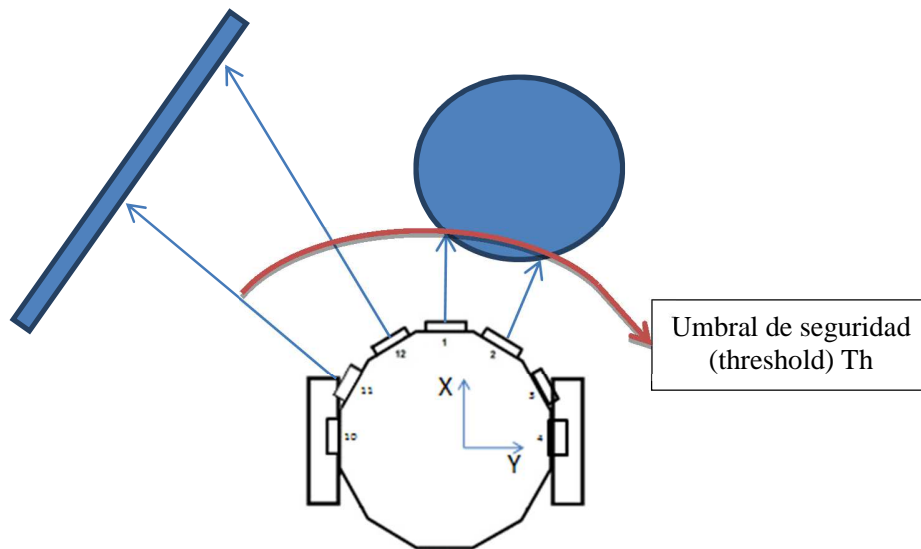


Figura 3.8. Emergency Behavior(EB).

El accionamiento del comportamiento de emergencia se acciona de acuerdo a las siguientes condiciones.

$$EB = \begin{cases} ON, & Sensorx < Th \\ OFF, & Sensorx \geq Th \end{cases}$$

Dónde: *Sensor x* es la lectura del sensor que detecta el objeto y *Th* es el umbral de detección que es propuesto por el programador del comportamiento.

3.1.6. Comportamiento 6: Vagabundeo (Wandering)

La figura 3.9 presenta el vagabundeo en el robot móvil. Esto significa que recorrerá la arena de exploración en búsqueda del objetivo sin trayectorias predefinidas (Wicaksono et al., 2009). Cuando el robot encuentra el objetivo u otra acción es requerida como buscar la meta, evasión de obstáculos, seguimientos de paredes o evadir estancamientos, el vagabundeo es inmediatamente anulado o subsumido por los demás comportamientos.

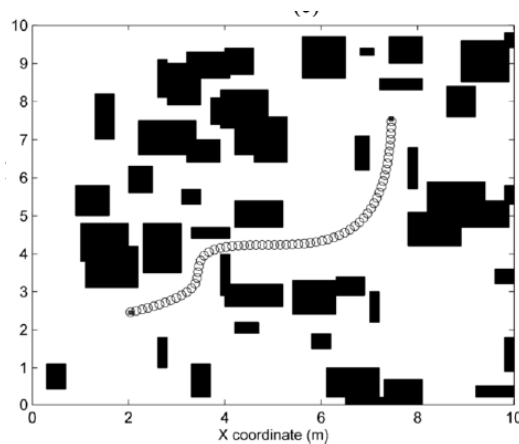


Figura 3.9. Wandering (Dongshu et al., 2011).

3.1.7. Comportamiento 7: Bloqueo de Motores (Locked-Rotor Detection Behavior)

Este comportamiento simplemente sirve para detectar si existe algún bloqueo en los motores por parte de algún objeto del entorno que navega, entonces el robot debe retroceder, luego evadir el obstáculo con alguna maniobra de giro y escapar del objeto que bloquea sus motores (Peng & Yumei, 2009).

En la documentación de estos comportamientos se encontró que existen también comportamientos emergentes los cuales son conductas que mezclan en un instante de tiempo y decisión más de un comportamiento de los vistos anteriormente para cumplir el objetivo. Las manifestaciones de varios comportamientos tratados como

un solo comportamientos son llamadas funcionalidad emergente, inteligencia emergente o inteligencia de enjambre. Esta clase de comportamiento está atribuido a la complejidad de la interacción entre los diferentes comportamientos diseñados. Entonces finalmente se le atribuye el apelativos de comportamientos emergente al cumplimiento de tareas donde varios comportamientos aportan para la búsqueda del objetivos utilizando un mecanismos naciente que no se contempló en el diseño inicial (Bräunl, 2008).

3.2. CONCLUSIONES DEL CAPITULO

Los comportamientos descritos anteriormente fue el resultado de una pesquisa exhaustiva en busca de describir los comportamientos más apropiados para ser implementados en el robot móvil *CARLITOS*. El conjunto descrito anteriormente puede garantizar una navegación segura del robot, pero implementar todo el conjunto sería una tarea compleja y laboriosa la cual se saldría de los lineamientos de esta investigación, y se debe tener en cuenta también que el microcontrolador de prestaciones medias que utiliza el robot requeriría una gran esfuerzo y no podría operar en óptimas condiciones.

Gracias a la literatura consultada podemos concluir que con la implementación de un conjunto base de comportamientos tales como:

- Seguimiento de pared derecha (Right-Wall Following)
- Seguimiento de pared izquierda (Left Wall Following)
- Evasión de obstáculos (Obstacle avoidance)

Se puede con total confianza realizar una navegación exitosa con el robot *CARLITOS* en entornos interiores de trabajos estáticos.

Capítulo 4.

Objetivo 2: Diseño de Comportamientos

En este capítulo se presenta el diseño de los tres comportamientos propuestos en las conclusiones de la sección anterior, que fueron implementados y utilizados en la navegación por el robot *CARLITOS*. Cada comportamiento se trató como un controlador independiente que posteriormente fueron integrados en el siguiente capítulo formándose así el esquema de control basado en comportamientos.

La técnica de inteligencia artificial seleccionada para el diseño del conjunto de comportamientos fue la lógica difusa. Con el uso de la lógica difusa se pudo hacer una abstracción intuitiva de los problemas asociados a cada comportamiento de una manera fácil, rápida y robusta atacando con gran fuerza los problemas de diseños implicados en cada comportamiento.

Para cada comportamiento tratado como un controlador independiente se realizó un análisis de las posibles variables de entrada que se necesitan, las variables de salida y acciones de control que se debería obtener de acuerdo al mecanismo de inferencia diseñado para cada caso.

En la construcción de los controladores se aplicó entonces el proceso habitual utilizando lógica difusa llevando las variables de entrada por el proceso de fusificación, para ser procesadas por un mecanismo de inferencia fundamentado en un *set* de reglas, que tomarán las decisiones que posteriormente serán defusificadas como valores reales para que los dispositivos electrónicos los interpreten y se ejecuten las acciones de control pertinentes.

La figura 4.1 muestra en general el proceso que conlleva el uso de la lógica difusa.

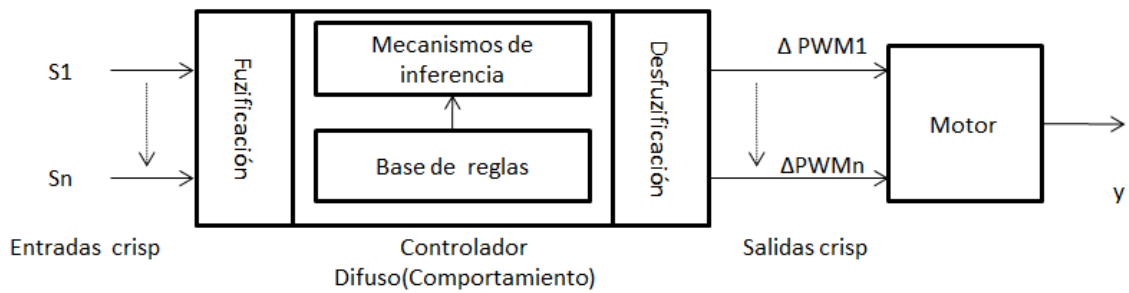


Figura 4.1. Proceso utilizado en el diseño de los comportamientos utilizando lógica difusa.

Como se puede ver en la figura 4.1 las variables de entrada son las lecturas de los sensores que se decidieron implementar para cada comportamiento y las variables de salida son diferenciales de PWM que cambiarán las revoluciones del motor para que cumpla el objetivo.

En la tabla 2 se presenta una lista con las variables de entrada que utilizará cada comportamiento y que variables de salida generara cada uno.

Entradas(Sensores)	Comportamientos (Controladores)	Salidas(Δ PWM)
eS4,deS4 (e=error; de=derivada)	Seguimiento de pared derecha	Δ PWM motor derecho
eS4,deS4 (e=error; de=derivada)	Seguimiento de pared izquierda	Δ PWM motor izquierdo
eS1, efD, efl (e=error; efD, efl=error frontal derecho e izquierdo)	Evasión de obstáculos	Δ PWM motor derecho e izquierdo

Tabla 2. Variables de entrada y salida propuestas para el conjunto de comportamientos diseñados.

Teniendo claro que variables de entrada se utilizaran para el diseño de los controladores propuestos, se inició estas labores con el seguimiento de pared derecha donde experimentalmente se diseñó un comportamiento totalmente reactivo utilizando el sensor S4 puesto para este comportamiento, en la sección 4.1 de este apartado se presenta la construcción de este comportamiento como inicio de las labores de diseño.

4.1. CONTROL POR DECISIONES ON - OFF

El diseño de este algoritmo de control se basó en una serie de reglas de decisión las cuales tratan de evitar la colisión del robot contra una pared variando las revoluciones del motor derecho. Estas variaciones efectúan el cambio de dirección del robot de una manera rudimentaria asemejándose un poco a un de control *On-Off* convencional. Dichos cambio que reorientar del autómata se realizan hacia una referencia (Set Point "SP") la cual es una distancia de 50 cm la cual debería mantener el robot con respecto al sensor lateral que utiliza.

La acción de control de este algoritmo se refleja en la toma del mando de uno de los efectores del robot para regular su velocidad (motor derecho). Las correcciones ejecutadas por este controlador son realizadas mediante un cambio de PWM (Pulse Width Modulation) que se aplica al motor derecho de la plataforma robótica.

Este cambio PWM se relacionó de acuerdo a una abstracción intuitiva del espacio, obteniéndose como resultado la variación de velocidad del motor y las respectivas correcciones de orientación para las diferentes zonas del entorno utilizado. La figura. 4.2 ilustra las zonas consideradas por el algoritmo de control.

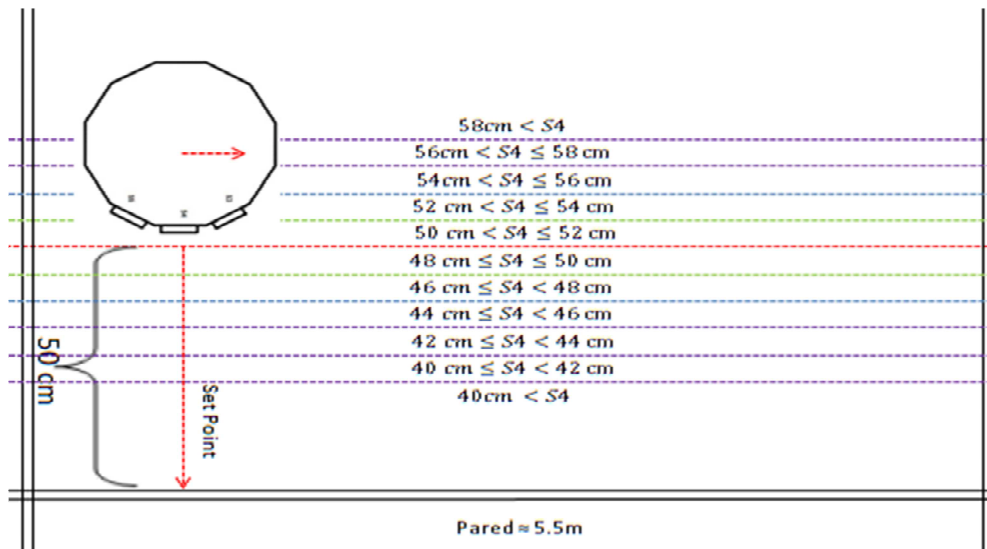


Figura 4.2. Zonas consideradas por el algoritmo de control.

Como se puede ver en la figura 4.2 las zonas comprenden sub-espacios en centímetro del espacio total de trabajo. El algoritmo de control evalúan entonces la medida actual que suministra el sensor S4 y de acuerdo a la zona en la que se encuentre el sensor se actualiza la carga numérica equivalente a un PWM constante al registro encargado de esta tarea en el microcontrolador, realizándose así correcciones de orientación.

La figura 4.3 presenta el diagrama de flujo en donde se aprecia la operación del algoritmo de control mencionado anteriormente.

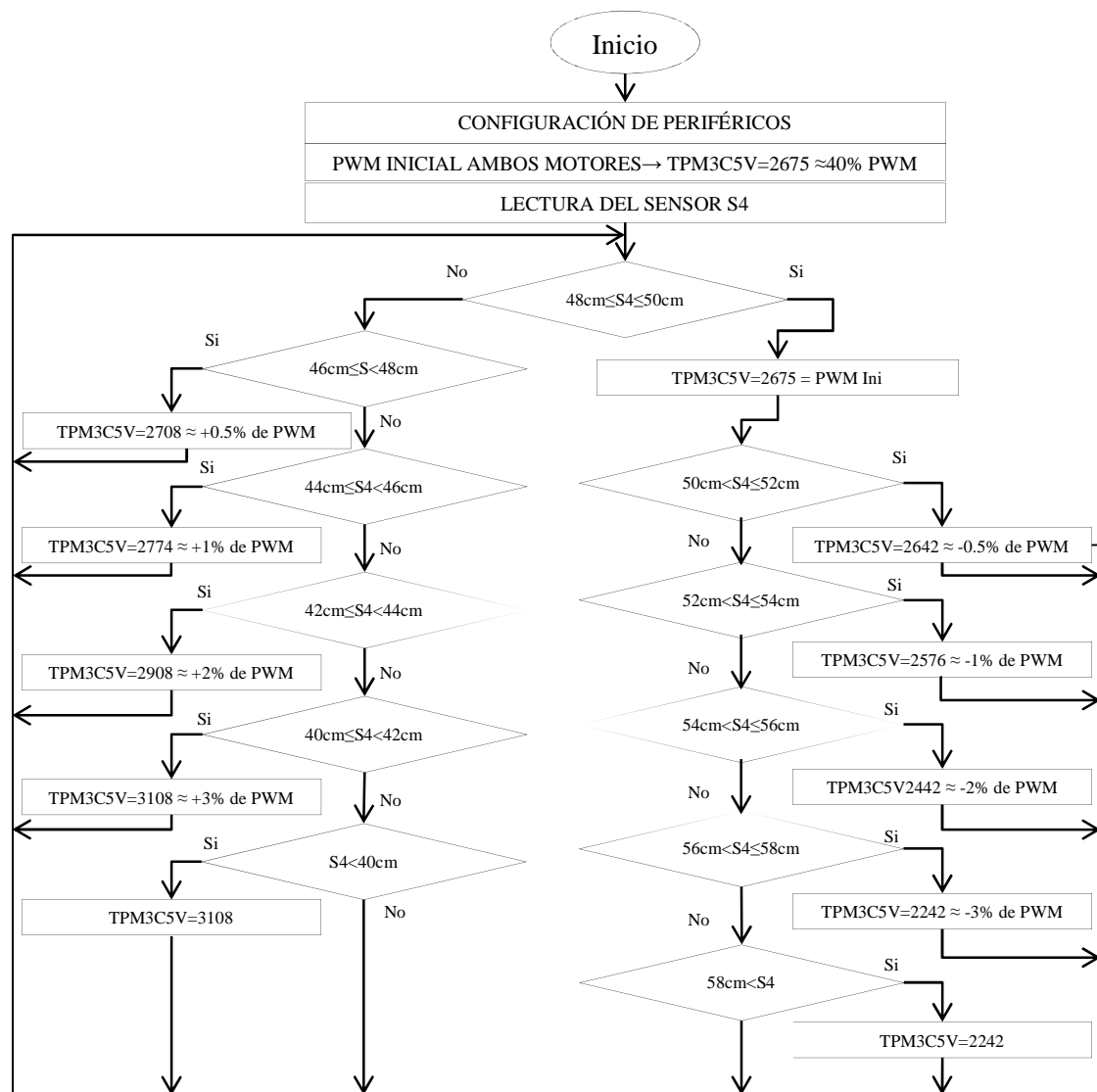


Figura 4.3. Diagrama de flujo – Algoritmo de control.

Funcionamiento del algoritmo:

- Una vez se configuran los periféricos necesarios para que el robot opere correctamente, se aplica una carga inicial de arranque de motores igual al 40% de PWM aproximada a 5.7 en los terminales de cada motor, lo cual es equivalente una carga numérica inicial al registro TPM3C5V y TPM3C2V de 2675 del microcontrolador.

- Las zonas describen un aumento o disminución en un porcentaje de PWM, esto se interpreta como un aumento o disminución en un porcentaje del PWM total al PWM actual que posee el motor derecho del robot de acuerdo a la próxima zona.
- Si el robot esta sobre la referencia, el PWM será igual al PWM inicial y se debe mantener constante para los dos motores.
- Si el robot está por debajo de la referencia (acercándose a la pared), el PWM en el motor derecho debe ser más grande que el de la zona anterior para aumentar la velocidad del motor y alejar al robot hacia la referencia.
- Si el robot está por encima de la referencia (alejándose de la pared), el PWM en el motor derecho debe ser menor al de la zona anterior para disminuir la velocidad del motor y acercar al robot a la referencia.
- El rango de navegación va desde 40cm hasta 58 cm.
- Si el robot esta por fuera del rango de navegación se dejará aplicado un PWM de saturación para corregir fuertemente la orientación del robot hacia la referencia para ambos casos.

4.1.1. Pruebas

Se realizaron 2 pruebas en un mismo escenario, se describieron las trayectorias de una forma experimental para analizar el comportamiento del robot, se verificó la operación de los algoritmos iniciales, y se analizaron los recorridos en busca de los máximos sobreimpulsos descritos en las trayectorias.

En la figura 4.4 se muestra el escenario real de navegación donde se pueden observar algunas trayectorias descritas por el robot.

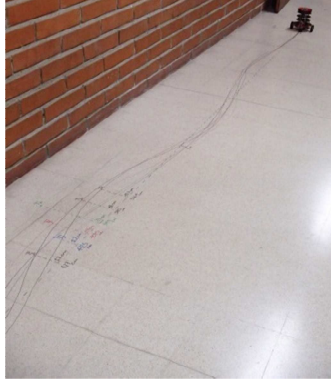


Figura 4.4. Escenario real.

4.1.1.1. Prueba 1

En esta prueba se implementó un algoritmo inicial que contempla las zonas y la estructura del escenario que se muestra en la figura 4.5. Esta prueba fue estructurada de la siguiente manera:

- Para cada recorrido el robot se posicionó inicialmente a 50 cm de distancia hacia la pared con respecto al sensor S4, el cual a su vez está encargado de tomar las mediciones hacia la pared derecha.
- Se realizaron un total de 6 intentos de navegación en los cuales se incluyen los recorridos erróneos y se documentaron solamente las navegaciones exitosas.
- Para cada recorrido se midió experimentalmente los sobreimpulsos máximos en centímetros con respecto a la horizontal que coincide con el centro del robot en su posición inicial y también se midió en qué parte del recorrido total se presentó dicho sobreimpulso.
- Se analizaron los comportamientos oscilatorios para cada recorrido.
- Se midió experimentalmente el error de llegada con respecto a la referencia del centro del robot para observar la corrección de llegada al final de cada recorrido.

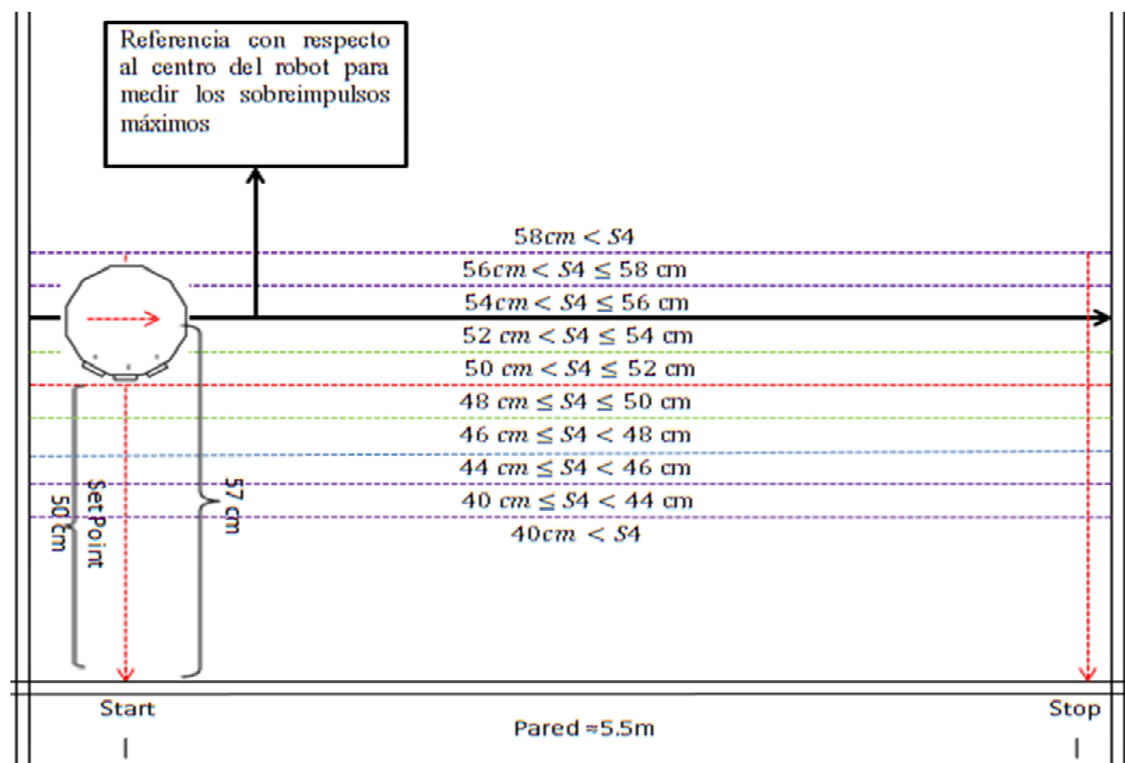


Figura 4.5. Escenario 1 de pruebas.

La tabla 3 presenta los recorridos fueron ejecutados y que sigue el siguiente orden:

Recorrido	Negro	rojo	verde	azul	Exitoso	No exitoso
1	X				X	
2	X					X
3		X			X	
4		X				X
5			X		X	
6				X	X	

Tabla 3. Ejecución de los recorridos prueba 1

4.1.1.2. Resultados prueba 1

En esta sección se muestran los registros fotográficos que se obtuvieron en esta prueba experimental. Cabe anotar que solo se registraron los recorridos que llegaron a la meta. La figura 4.6 muestra el inicio del escenario o etiqueta de inicio de recorrido (*Start*)

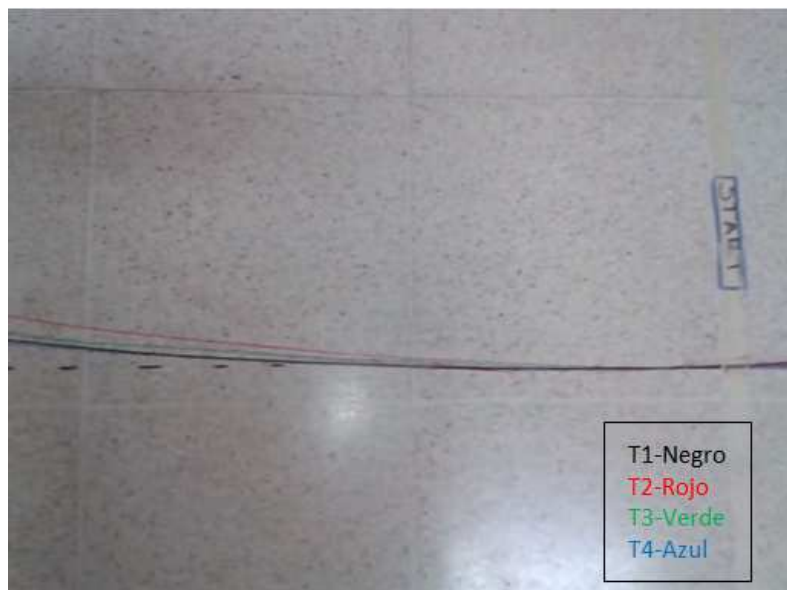


Figura 4.6. Inicio de recorrido, T(1-4)“Trayectorias”.

Se puede observar que al iniciar la prueba, los recorridos que describe el robot son trayectorias parecidas para los cuatro intentos de navegación exitosos registrados. Este comportamiento inicial se debe a las condiciones controladas de comienzo de la navegación y a que las perturbaciones en estos momentos iniciales no son críticas. Una vez transcurrida una navegación aproximada a los 2 m se evidencia los comportamientos oscilatorios del robot en busca de las correcciones hacia el Set Point (SP). La figura 4.7 presenta las oscilaciones iniciales de recorrido.

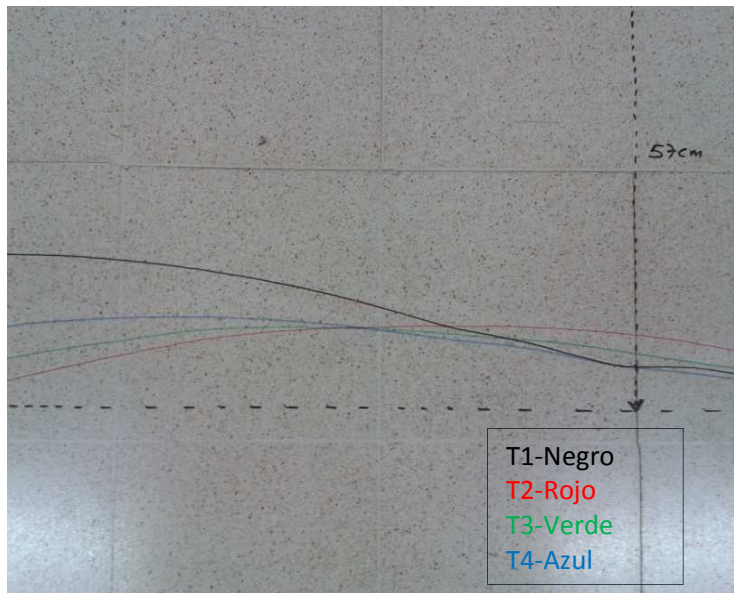


Figura 4.7. Oscilaciones iniciales.

En las cuatro trayectorias descritas por el robot se pudo observar que los mayores sobreimpulso se manifestaron siempre finalizado el recorrido (últimos 2 metros). La figura 4.8 muestra la zona de los sobreimpulsos máximos.

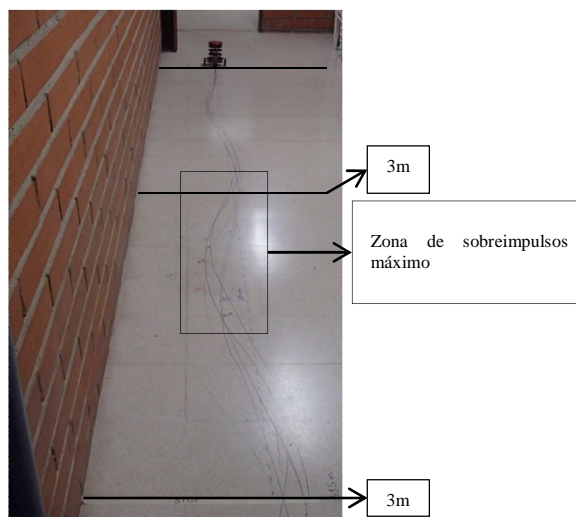


Figura 4.8. Sobreimpulso.

Para las mediciones que se presentan en la figura 4.9 se tuvo como referencia la horizontal central inicial de robot. Se analizó visualmente donde se presentaron los sobreimpulsos máximos y experimentalmente con ayuda de una cinta métrica se tomaron las medidas de interés.

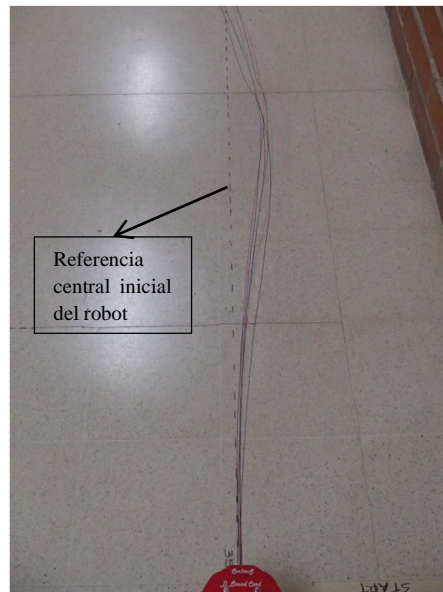


Figura 4.9. Referencia.

La figura 4.10 muestra las mediciones de los sobreimpulsos realizadas en esta zona.

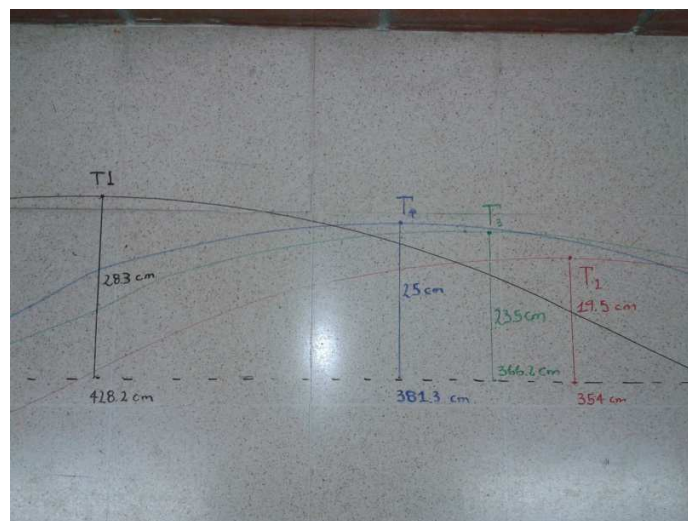


Figura 4.10. Mediciones de sobreimpulsos.

En la tabla 4 se puede ver que los sobreimpulsos máximos fueron descritos por el robot en una misma zona en el espacio de las navegaciones. Estos están asociados a la acción de corrección fuerte que ejecuta el algoritmo de control tratando de minimizar el error que se va acumulando. Esta acumulación de error se puede interpretar como los desfases de orientación del robot con respecto a la referencia central del robot.

Trayectoria (T)	Sobreimpulso (cm)	Distancia recorrida (cm)
T_1	28.3	428.2
T_2	19.5	354
T_3	23.5	366.2
T_4	25	381.3

Tabla 4. Mediciones de sobreimpulso.

Al culminarse los recorridos se midieron los errores de orientación final con el cual finalizaron las trayectorias, se pueden evidenciar estas mediciones en la figura 4.11.

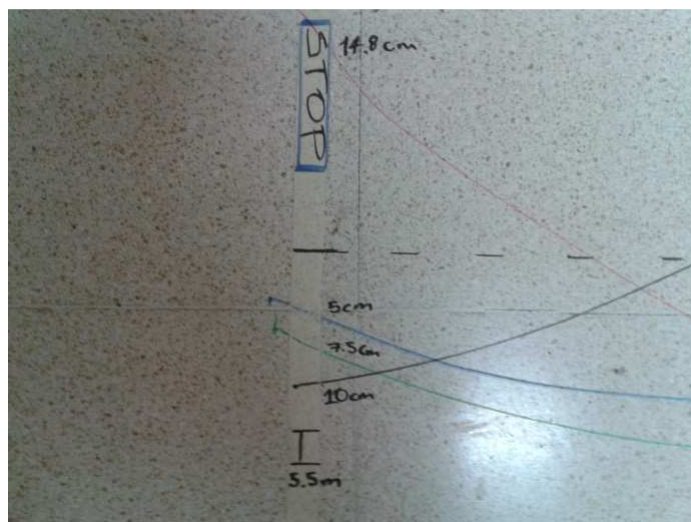


Figura 4.11. Error de orientación.

Trayectoria (T)	Error de orientación (cm)
T_1	10
T_2	14.8
T_3	7.5
T_4	5

Tabla 5. Error de orientación.

Se infiere de la figura 4.11 que, basándose en el segmento final de las trayectorias y los sobreimpulsos máximos registrados, las trayectorias que describieron un mejor desempeño del algoritmo de control fueron T_3 y T_4 . En la tabla 5 se observa que estos dos recorridos presentaron errores finales menores que los demás y sobreimpulsos parecidos.

4.1.1.3. Prueba 2

En esta prueba se implementó una mejora del algoritmo de control la cual surgió de un mejor análisis intuitivo del espacio utilizado en la prueba anterior.

Se ajustaron casi los mismos lineamientos descritos en la prueba anterior agregándose un recorrido más, y se hizo la inclusión de otras zonas como lo muestra el escenario descrito en la figura 4.12.

La tabla 6 muestra el orden de ejecución de las pruebas.

La figura 4.13 dos imágenes de diferentes ángulos donde se presentan las trayectorias descritas por el robot en la prueba numero dos.

Recorrido	Negro	rojo	verde	azul	Exitoso	No exitoso
1	X				X	
2		X				X
3		X			X	
4			X		X	
5				X	X	
6	X				X	
7	X				X	

Tabla 6. Ejecución de los recorridos prueba 2.

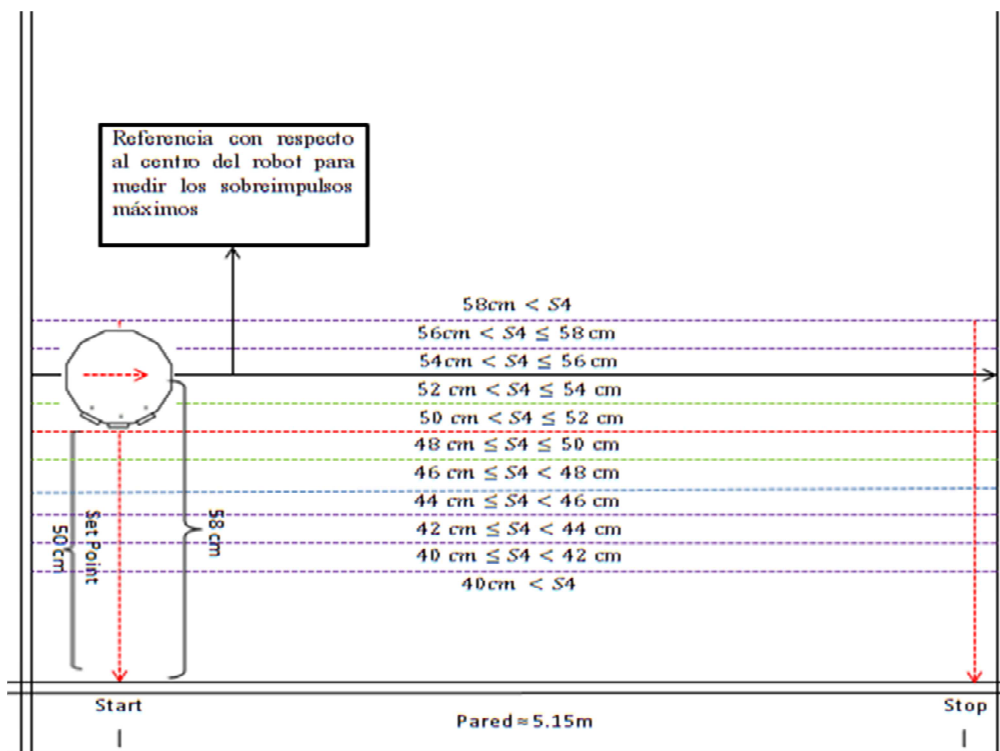


Figura 4.12. Escenario 2 de pruebas.

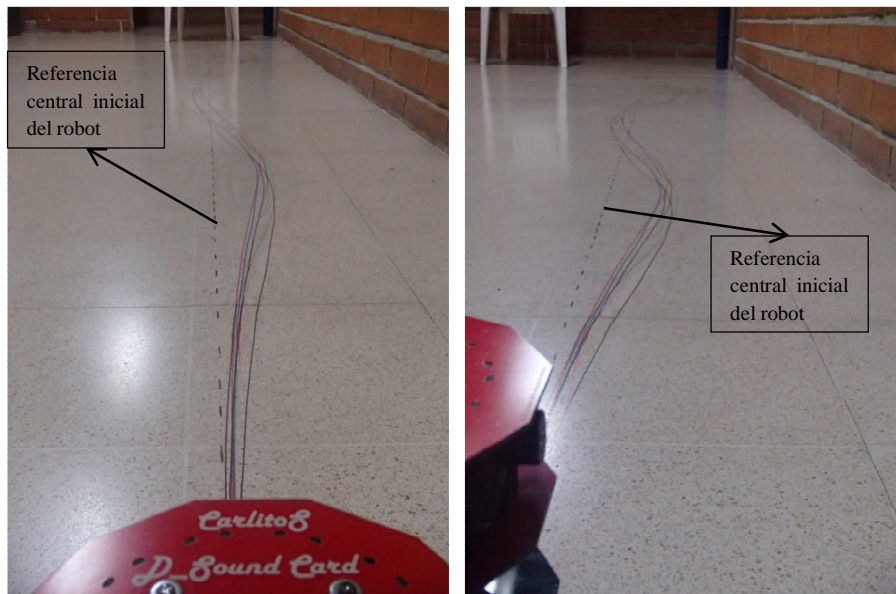


Figura 4.13. Inicio de recorrido, T(1-4)“Trayectorias” y oscilaciones iniciales.

4.1.1.4. Resultados prueba 2

Para esta prueba también se aplicó el mismo procedimiento para realizar las mediciones de sobreimpulsos máximos y el error de orientación al final de la trayectoria. En las figura 4.14 y 4.15 se presentan las mediciones realizadas.

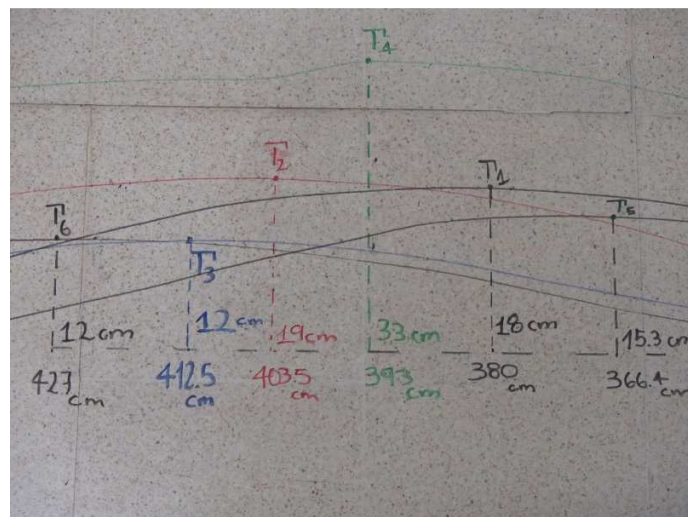


Figura 4.14. Mediciones de sobreimpulsos.

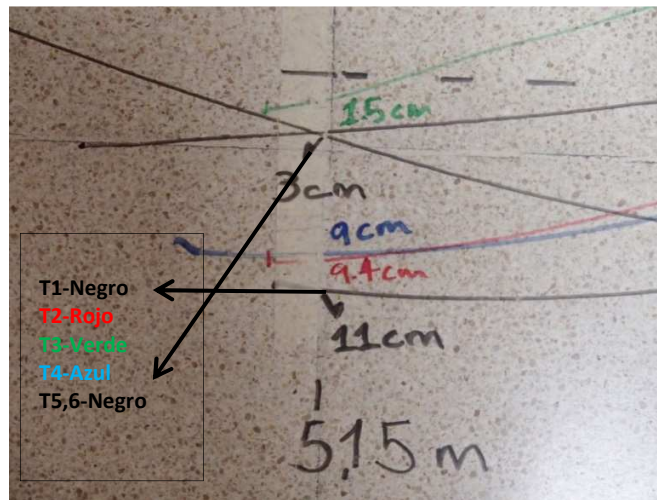


Figura 4.15. Error de orientación.

De la Tabla 7 se puede ver que los sobreimpulsos medido fueron menores a los de la prueba anterior.

Trayectoria (T)	Sobreimpulso (cm)	Distancia recorrida (cm)
T_1	18	380
T_2	19	403.5
T_3	33	393
T_4	12	412.5
T_5	15.3	366.4
T_6	12	427

Tabla 7. Mediciones de sobreimpulso

Esto indica que las mejoras al algoritmo de control surtieron efectos positivos en la navegación del robot. Comparando entonces directamente los resultados de la Tabla 4 y 7, en la segunda prueba las magnitudes de los sobreimpulsos fueron menores siendo estos inferiores a los 20 cm en su mayoría, lo cual muestra un mejor desempeño en las navegaciones, de lo anterior se excluye la trayectoria T_3 que fue

la trayectoria que tuvo un sobreimpulso mayor a nivel global de las pruebas. Esta trayectoria mencionada al igual que la T_1 de la prueba 1 son las que presentaron los estados más críticos en esta medición. Estos dos comportamientos se pueden asociar a reflexiones especulares donde la onda de ultrasonido que choca la pared, nunca pudo retornar al receptor del sensor generándose entonces medidas espurias. Estas medidas espurias permiten al robot hacer inmersiones fuertes hacia la pared o alejarse mucho de ella de acuerdo a la zona donde navega el robot, por consecuencia se describen oscilaciones muy fuertes que son análogas a sobreimpulsos de mayor magnitud. También las oscilaciones más fuertes se pudieron generar por la mala posición del robot en su localización inicial la fue controlada por el programador de la prueba.

Se ratifica con las medidas finales de navegación de esta prueba que el algoritmo arrojó mejores resultados por medio de la corrección de error que se presentó al final de la prueba, se analizó que aunque la trayectoria T_3 fue la más crítica, realizó las correcciones necesarias para tratar de permanecer en la referencia como se observó, al igual que las demás. Se puede notar también en la figura 4.15 que la acción de control hizo que el robot terminara todos los recorridos tratando de suprimir los errores por encima de la referencia. Esto data que las acciones de control fueron iguales en cierto grado. En la tabla 8 se presentan los errores de orientación final para cada recorrido de la prueba dos.

Trayectoria (T)	Error de orientación (cm)
T_1	11
T_2	9.4
T_3	15
T_4	9
T_5	3
T_6	3

Tabla 8. Error de orientación.

4.1.2. Conclusiones de las Pruebas

- Se puede inferir entonces que las dos pruebas se consideraron exitosas puesto que la mayoría de las navegaciones culminaron todo el recorrido.
- Como se mencionó en el encabezado de esta sesión el algoritmo de control está basado en una serie de decisiones aunque no es una técnica de control formal se realizaron buenas correcciones en el control en general para los recorridos.
- Se pudo analizar que al hacer subdivisiones más finas del entorno de trabajo, aumentan las decisiones del algoritmo de control y por ende se realizan mejores trayectorias de navegación, lo cual se ve reflejado en la reducción de los sobreimpulsos y un error de llegada de menor magnitud, pero por otra parte se produjo un aumento en las líneas de programación que toma estas decisiones, volviendo el algoritmo más extenso y demandando más trabajo al microcontrolador.
- Aunque las medidas que se realizaron de una forma experimental, se pudo constatar que la mejora del algoritmo (mas subdivisiones del entorno de navegación) arrojó mejores resultados frente a la prueba uno.

4.1.3. Premisa

El razonamiento realizado en las pruebas anteriores sirvió como punto de partida para iniciar el diseño de los controladores basado en comportamiento que utiliza el robot para su navegación.

4.2. DISEÑO DE COMPORTAMIENTO DE SEGUIMIENTO DE PARED DERECHA

La naturaleza de este comportamiento como su nombre lo indica es el seguimiento de paredes situadas a la derecha del robot. El seguimiento de esta pared se realizó

por medio de las mediciones suministradas por el sensor S4 que incluye el módulo de precepción que posee el robot *CARLITOS*.

El diseño de este comportamiento inicial está inspirado en las pruebas del comportamiento inicial trazado en la sección anterior. De ahí se recopilieron múltiples experiencias tales como el apropiado manejo de los sensores, interpretación de las lecturas adquiridas por el sensor y manejo de los motores, programación para manejo de los periféricos del microcontrolador entre otras. Estos aportes incidieron fuertemente a la construcción de este comportamiento.

En este comportamiento el robot detecta a su costado derecho las paredes u objetos tratando de seguir mayormente las paredes. Como referencia hacia los posibles elementos y paredes que el robot pudiese encontrar a su costado derecho, se planteó un límite de permanencia hacia estos de 60 cm para efectos de pruebas. Este límite de permanencia es conocido en teoría de control como *Set-point* o punto de ajuste en español, que es una distancia que deberá mantener el robot hacia los elementos o paredes encontrados a su lado derecho. En la figura 4.16 se muestra idealmente como el robot toma las lecturas de las paredes derechas en un escenario de pruebas propuestos.

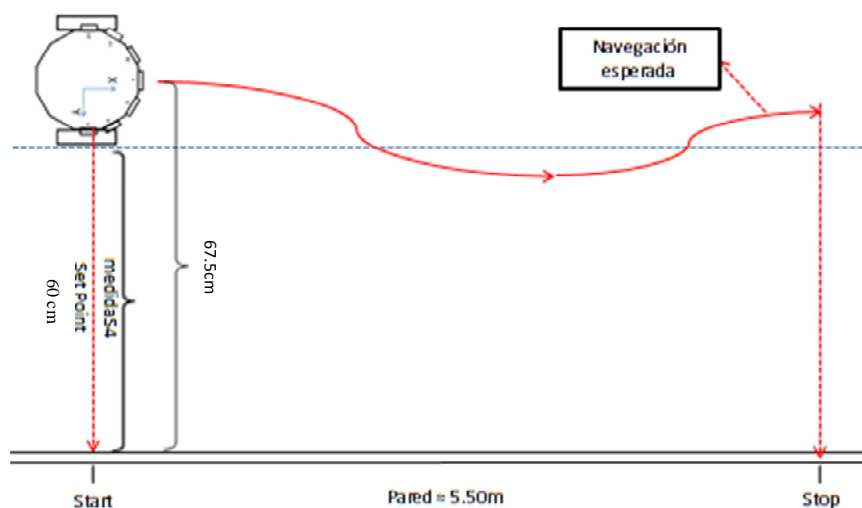


Figura 4.16. Seguimiento de pared derecha

El escenario propuesto de navegación es una pared ubicada a la derecha del robot en la cual se delimito un espacio aproximado de 5.50 m para pruebas. En la figura 4.16 se observa las referencias de medida de sensor S4, la referencia del centro del robot hacia la pared y cómo sería la navegación esperada.

4.2.1. Diseño

En el diseño se realizó el tratamiento de la variable de medición que suministra el sensor S4 respecto al *Set-point* propuesto de 50 cm hacia la pared derecha. De acuerdo a la relación *Set-point*-variable se trabajó con la de error de medición de S4 que es la diferencia que existe entre la referencia y el estado actual de la variable en un momento en la navegación. La relación se puede expresar de la siguiente manera:

$$eS4 = Setpoint - medidaS4$$

Dónde:

- *Setpoint* = 60cm (referencia propuesta).
- *medidaS4*, medida actual del sensor 4.

Como al ser la variable error una diferencia entre la referencia y la variable medida. Esta puede tomar valores positivo, negativos o cero, lo cual nos ayudara a proponer las decisiones que deberá tomar el robot en este diseño.

Se infiere entonces lo siguiente:

$$eS4 = \begin{cases} +; si Setpoint > medidaS4 \\ 0; si Setpoint = medidaS4 \\ -; si Setpoint < medidaS4 \end{cases}$$

El signo del resultado de la variable de error se puede interpretar como lo siguiente:

- +: El robot está cercano a la pared derecha.
- 0: El robot está manteniendo la distancia de set-point hacia la pared.
- -: El robot está alejado de la pared derecha.

También se trabajó con la derivada del error el cual se aproxima como una operación de primera diferencia, la cual se expresa como una diferencia entre errores en dos instantes de muestreo.

$$deS4 = eS4_{(k)} - eS4_{(k-1)}$$

Al aproximarse la derivada del error como una primera diferencia, esta también puede tomar valores positivos, cero o negativos como se muestra en la siguiente expresión.

$$deS4 = \begin{cases} +; \text{Derivada creciente} \\ 0; \text{Errores constante} \\ -; \text{Derivada decreciente} \end{cases}$$

Con la derivada podemos saber si el robot se está alejando, acercando o igual a la referencia y dependerá del cuadrante en el que analice. La figura 4.17 muestra el análisis completo de la derivada del error y el error.

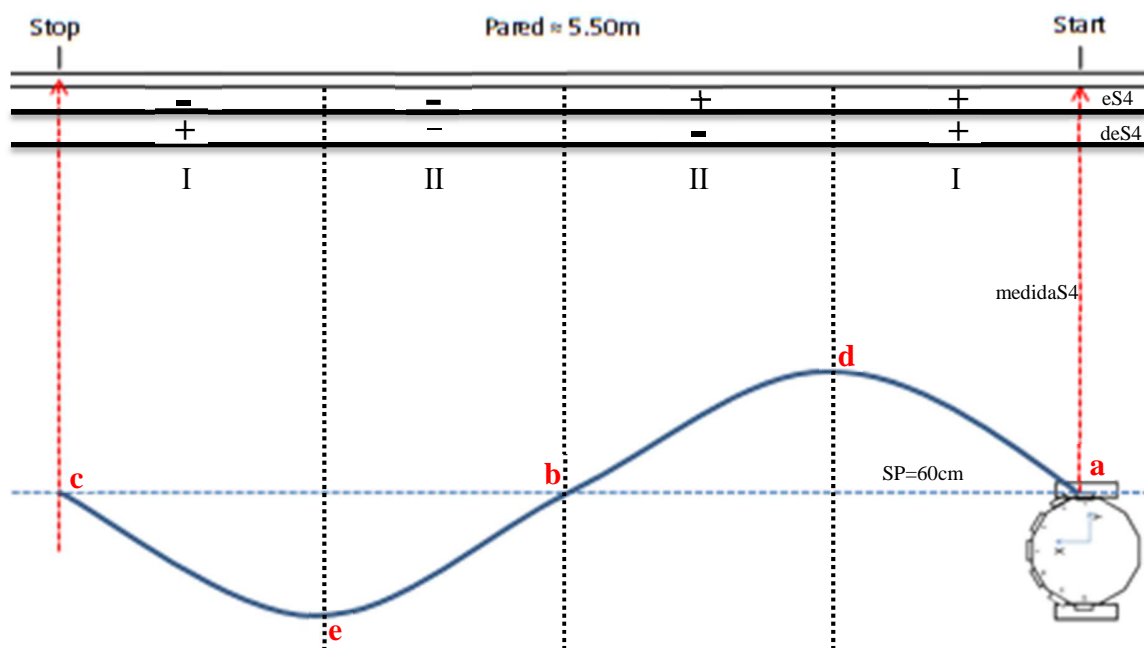


Figura 4.17. Análisis del error y la derivada del error de medida de S4.

De donde se extrajo la tabla 9 que sirvió para el diseño del controlador difuso de seguimiento de pared derecha.

Regla N°	eS4	deS4($\Delta eS4$)	Acción de Control	Valor de la referencia
1	+	+	DPWMDW= \uparrow	I
2	+	–	DPWMDW= \uparrow	II
3	–	–	DPWMDW= \downarrow	III
4	–	+	DPWMDW= \downarrow	IV
5	0	0	DPWMDW=0	a
6	0	–	DPWMDW= \downarrow	b
7	0	+	DPWMDW= \uparrow	c
8	+	0	DPWMDW= \uparrow	d
9	–	0	DPWMDW= \downarrow	e

Tabla 9. Análisis del comportamiento-Seguimiento de pared derecha.

La tabla comprende los posibles panoramas que se puede presentar en la navegación con este comportamiento a través del análisis del error y la derivada del error. Como acción de control tenemos entonces que se aumentará o disminuirá un porcentaje de PWM al porcentaje inicial de arranque del motor derecho, este cambio de porcentaje es llamado DPWMD (Delta de PWM Derecho) el cual aumentará o disminuirá las revoluciones del motor derecho para realizar las correcciones de navegación de la plataforma robótica respecto a las paredes a la derechas. En la figura 4.18 se aprecia el diagrama de control donde se puede ver claramente cómo se constituye el controlador difuso de seguimiento de pared derecha

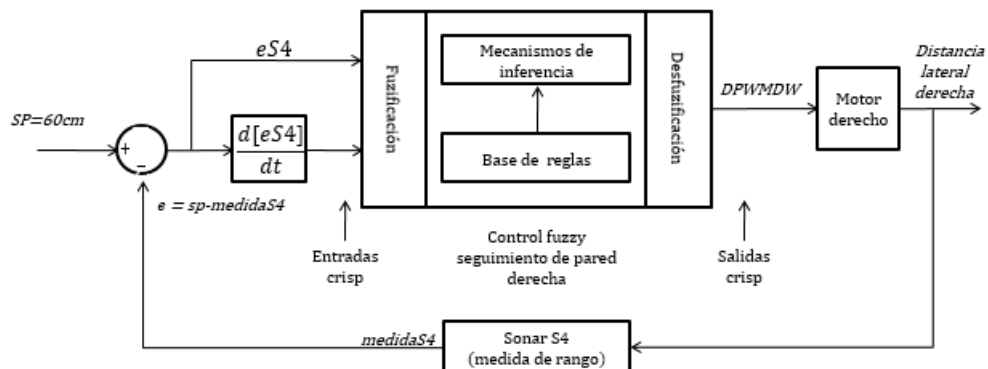


Figura 4.18. Diagrama de control *fuzzy* seguimiento de pared derecha.

La figura 4.19 presenta una perspectiva con el robot donde se tiene lo siguiente:

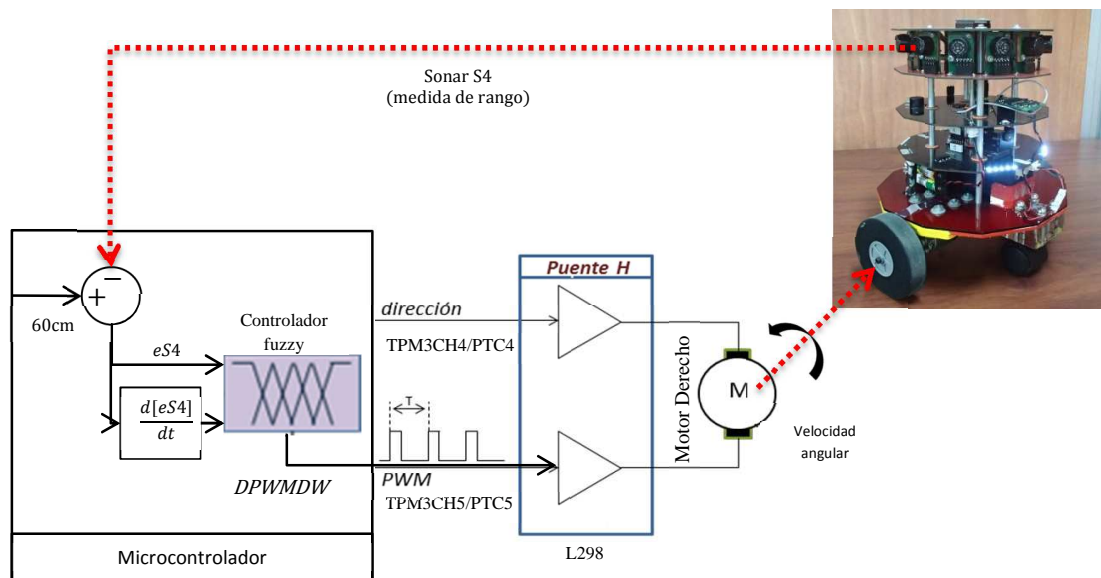


Figura 4.19. Estructura de control del seguimiento de pared derecha.

4.2.2. Controlador difuso

4.2.2.1. Caracterización de variables lingüísticas para el comportamiento seguimiento de pared derecho

La tabla 10 muestra la caracterización de las variables de error, la tabla 11 la derivada del error y la tabla 12 el delta de PWM para el motor derecho tenidas en cuenta para este comportamiento, dichas tablas presentan el tratamiento de las variables como conjuntos difusos.

CARACTERIZACIÓN DE LA VARIABLE LINGÜÍSTICA DEL ERROR DEL SENSOR S4		
Nombre: eS4		
Tipo: entrada trapmf	Tipo: entrada trimf	Tipo: entrada trapmf
Conjunto de términos: eS4_Negativo. (eS4_Nt)	Conjunto de términos: eS4_zero. (eS4_Zr)	Conjunto de términos: eS4_Positivo. (eS4_Pt)
Universo de discusión: U=[-20 20]	Universo de discusión: U=[-20 20]	Universo de discusión: U=[-20 20]
Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: eS4_Nt = [-30 -20 -10 0]	Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: eS4_Zr = [-10 0 5]	Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: eS4_Pt = [0 5 20 30]

Tabla 10. Caracterización variable de error seguimiento de pared derecha.

CARACTERIZACIÓN DE LA VARIABLE LINGÜÍSTICA DE LA DERIVADA DEL ERROR DEL SENSOR deS4		
Nombre: deS4		
Tipo: entrada trapmf	Tipo: entrada trimf	Tipo: entrada trapmf
Conjunto de términos: Derivada del deS4_Negativo. (deS4_Nt)	Conjunto de términos: Derivada del deS4_zero. (deS4_Zr)	Conjunto de términos: Derivada del deS4_Positivo. (deS4_Pt)
Universo de discusión: U=[-20 20]	Universo de discusión: U=[-20 20]	Universo de discusión: U=[-20 20]
Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: deS4_Nt = [-30 -20 -10 0]	Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: deS4_Zr = [-10 0 5]	Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: deS4_Pt = [0 5 20 30]

Tabla 11. Caracterización variable derivada del error del seguimiento de pared derecha.

CARACTERIZACIÓN DE LA VARIABLE LINGÜÍSTICA DE LA SALIDA DELTA DE PWM MOTOR DERECHO				
Nombre: DPWMDW				
Tipo: Salida trimf	Tipo: Salida trimf	Tipo: Salida trimf	Tipo: Salida trimf	Tipo: Salida trimf
Conjunto de términos: Delta de PWM Derecho Negativo Gande (DPWMDW_NG)	Conjunto de términos: Delta de PWM Derecho Negativo Pequeño (DPWMDW_NP)	Conjunto de términos: Delta de PWM Derecho Zero (DPWMDW_Zr)	Conjunto de términos: Delta de PWM Derecho Positivo Pequeño (DPWMDW_PP)	Conjunto de términos: Delta de PWM Derecho Positivo Gande (DPWMDW_PG)
Universo de discusión: U=[-350 350]	Universo de discusión: U=[-350 350]	Universo de discusión: U=[-350 350]	Universo de discusión: U=[-350 350]	Universo de discusión: U=[-350 350]
Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: DPWMD_NG = [-20 -10 0]	Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: DPWMD_NP = [-15 -5 5]	Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: DPWMD_Zr = [190 200 210]	Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: DPWMD_PP = [580 590 600]	Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: DPWMD_PG = [290 300 310]

Tabla 12. Caracterización variable de salida de DPWMD.

La salida de PWM tanto para el motor derecho como para el motor izquierdo pueden tomar valores en un rango de [0 - 6687] equivalente a [0-12 voltios]. El registro encargado de controlar el periférico de PWM del robot admite cambios en el rango de [0 - 6687], siendo '0' análogo a '0% de PWM o cero voltios' y '6687' análogo a '100% de PWM o 12 voltios'. Si sumamos o restamos a un PWM inicial diferenciales de PWM para que el registro varíe en su rango, tendremos como consecuencia un aumento o decremento del PWM en los motores. Estas variaciones serán interpretadas por el motor como aumento o disminución de voltajes en sus terminales respectivamente. En ese orden de ideas, podemos entonces hablar entonces de cambios positivos o negativos en el PWM que se verán reflejado como aumento o disminución de las velocidades de los motores del robot.

Teniendo entonces bien definidas las variables de entrada (eS4 y deS4) y la variable de salida (DPWMD), podemos realizar la construcción de la tabla FAM (Fuzzy

Associative Memory) la cual hace inferencia de los conjuntos difusos de las variables de entrada y los conjuntos de las variables de salida según el análisis de la figura 4.17.

En la tabla 13 se presenta la tabla FAM la cual presenta la inferencia realizada para el comportamiento de pared derecha.

4.2.2.2. Tabla FAM

eS4 \ deS4	deS4_Nt	deS4_Zr	deS4_Pt
eS4_Nt	DPWMDW_NP	DPWMDW_NG	DPWMDW_NP
eS4_Zr	DPWMDW_NP	DPWMDW_ZR	DPWMDW_PP
eS4_Pt	DPWMDW_PP	DPWMDW_PG	DPWMDW_PP

Tabla 13. Tabla FAM seguimiento de pared derecha.

De la tabla 13 se extraen el conjunto de reglas que servirá como reglas para mecanismo de inferencias del controlador. Dichas reglas fusionan las variables semánticas de los conjuntos difusos de las variables de entrada y las variables de salida mediante sentencias If-Then que es el método del diseño propio de la lógica difusa. Si bien este controlador de seguimiento de se compone como un sistema MISO (Multiple Input, Single output)(ver figura 4.18) las reglas If-Then presentaran la misma características de estos sistemas MISO como se muestra en el siguiente ejemplo:

Para el resultado señalado en verde en la tabla 13, la construcción de la regla se puede definir con la siguiente sintaxis:

- If (eS4 is eS4-Nt) and (deS4 is deS4-Nt) then (DPWMD is DPWMD-NP) (1)

Teniendo clara la sintaxis de las reglas, se presentan las nueve reglas que componen este comportamiento.

- 1. If (eS4 is eS4-Nt) and (deS4 is deS4-Nt) then (DPWMDW is DPWMDW-NP)
- 2. If (eS4 is eS4-Nt) and (deS4 is deS4-Zr) then (DPWMDW is DPWMDW-NG)

- 3. If (eS4 is eS4-Nt) and (deS4 is deS4-Pt) then (DPWMDW is DPWMDW-NP)
- 4. If (eS4 is eS4-Zr) and (deS4 is deS4-Nt) then (DPWMDW is DPWMDW-NP)
- 5. If (eS4 is eS4-Zr) and (deS4 is deS4-Zr) then (DPWMDW is DPWMDW-Zr)
- 6. If (eS4 is eS4-Zr) and (deS4 is deS4-Pt) then (DPWMDW is DPWMDW-PP)
- 7. If (eS4 is eS4-Pt) and (deS4 is deS4-Nt) then (DPWMDW is DPWMDW-PP)
- 8. If (eS4 is eS4-Pt) and (deS4 is deS4-Zr) then (DPWMDW is DPWMDW-PG)
- 9. If (eS4 is eS4-Pt) and (deS4 is deS4-Pt) then (DPWMDW is DPWMDW-PP)

4.2.2.3. Simulación del comportamiento

De la figura 4.20 podemos ver entonces que con las particiones realizadas para las variables lingüísticas se obtuvo una superficie de control que tiende a comportarse como un plano, lo cual quiere decir que las acciones de control tienden a una relación lineal entre las entradas y salidas del sistema.

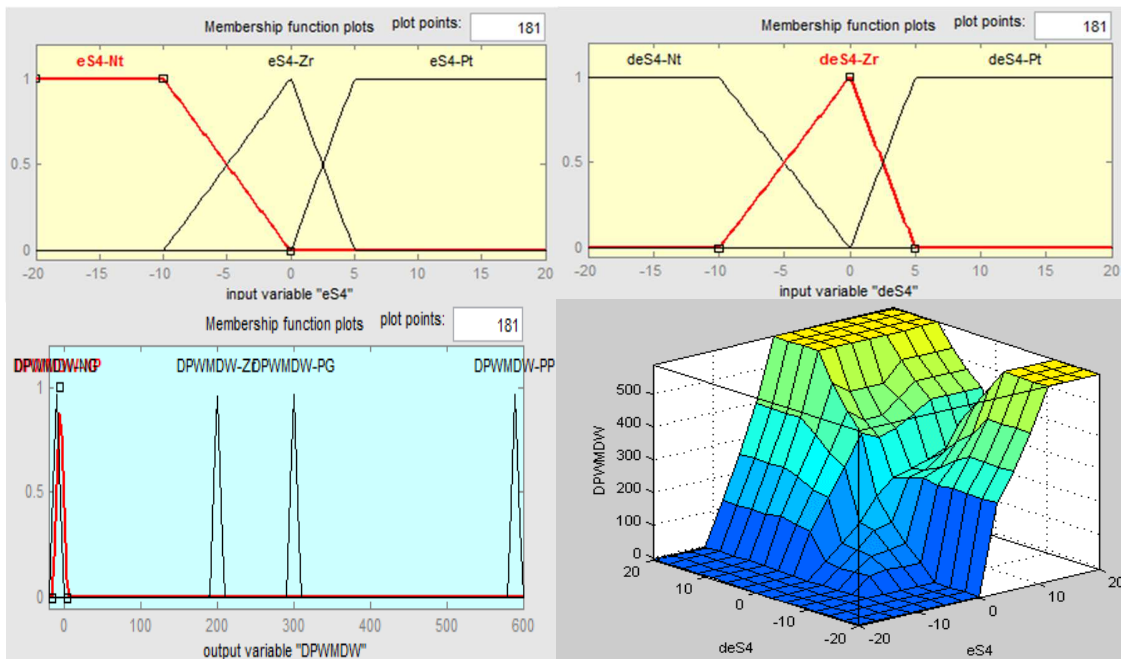


Figura 4.20. Partición de variables lingüísticas de entrada en conjuntos difusos, salida PWMD y superficie de control para el comportamiento de seguimiento de pared derecha.

Se observa también que los conjuntos difusos de las variables lingüísticas tanto de entrada como de salida son asimétricos debido a los ajustes realizados y la experimentación exhaustiva realizada en las pruebas de chequeo de este comportamiento en la plataforma robótica, dichos ajustes son necesarios para mejorar las acciones de control. Estas asimetrías se pueden asociar a los dispositivos utilizados que aunque idealmente tienen las mismas propiedades físicas y de construcción, estos operan de maneras diferentes en la realidad.

Se puede apreciar también en la figura 4.20 que las funciones de la variable salida PWM son *singleton* los cuales son los valores de cada conjunto difuso de salida con mayor valor de pertenencia. La defusificación se realizó utilizando el método del centroide o centro de área, este método de defusificación fue seleccionado por su fácil interpretación e implementación en la codificación de las funciones en el microcontrolador. Cabe recordar que el proceso de defusificación en lógica difusa sirve para que el valor resultante de la inferencia realizada por el mecanismo utilizado (valor de control), pueda ser interpretado por los efectores. Para este caso sería volver un valor de pertenencia difuso de 0 a 1 a diferenciales de PWM para ser aplicado a los motores.

4.3. DISEÑO DE COMPORTAMIENTO EVASIÓN DE OBSTÁCULOS

Este comportamiento fue diseñado para la evasión de obstáculos que se puedan ubicar en las zonas de detección frontal, frontal izquierda y frontal derecha del robot. Los dispositivos que están a cargo de la detección de objetos son los sensores S1, S2, S3, S11 y S12, donde S1 está encargado de la zona frontal, S11 y S12 están encargados de la zona frontal izquierda, y S2 y S3 de la zona frontal derecha. En la figura 4.21 se muestran los sensores frontales utilizados para el diseño de este comportamiento.



Figura 4.21. Sensores frontales.

Este comportamiento posee la característica hacer girar el robot a la derecha o hacia la izquierda dependiendo de la ubicación del objeto presente en la zona de detección. La capacidad de giro a ambos lados implica el control de los dos motores, lo cual aumenta la complejidad del controlador a diferencia de los seguimientos de paredes donde regularmente se controla un solo dispositivo de esta clase.

Otra característica que presenta este comportamiento es que en algunas situaciones, el giro por defecto es hacia la derecha debido a la inferencia sensorial realizada y por seguridad para que el robot no sufra ningún daño. En general la dinámica del comportamiento es la siguiente.

- Cuando se presenta un objeto a la izquierda, el PWM del motor izquierdo (PWMI) aumenta por medio de un diferencial (DPWMI) para evadir el objeto hacia la derecha.
- Cuando se presenta un objeto a la derecha, el PWM del motor derecho (PWMD) aumenta por medio de un diferencial (DPWMD) para evadir el objeto hacia la izquierda.
- Cuando se presenta un objeto en frente del robot, el PWM del motor derecho (PWMD) aumenta por medio de un diferencial (DPWMD) para evadir el objeto por defecto hacia la derecha.
- Cuando se presentan obstáculo a ambos lados encerrando al robot, este aumenta el PWM del motor derecho (PWMD) y disminuye el PWM del motor izquierdo (PWMI) para evitar el estancamiento en zonas cerradas, efectuando un giro a la izquierda de 90 grados.

4.3.1. Diseño

Para el diseño de este control como se mencionó anteriormente, se hizo uso de los sensores frontales del robot S1, S2, S3, S12 y S12. Se infiere entonces que con estos sensores se debe delimitar una zona de detección en la cual podrá percibir los objetos circundantes a la zona frontal del robot.

El límite de detección o *set-point* para los sensores se propuso de 75 cm hacia objetos frontales laterales y a 120 a objetos frontales para efectos de pruebas. Esto quiere decir que si un objeto esta por fuera de esta zona de detección, no se efectuarán acciones de control o por el contrario, si existe alguna influencia de algún objeto en la zona de detección o en sus límites, se realizaran las acciones de control correspondientes.

Para la detección de objetos frontales izquierdos, se utilizó las lecturas suministradas por los sensores S11 y S12. Para saber que tan cerca o lejano está el objeto y discriminar la posición del objeto, la ecuación de los errores de distancias propuesta se establece de la siguiente manera.

$$efI = Setpoint - \min(medidaS12, medidaS12)$$

Dónde:

- efI = Error frontal izquierdo
- $Setpoint$ = 75 cm (referencia propuesta).
- $medidaS11$, medida actual del sensor 11.
- $medidaS12$, medida actual del sensor 12.
- *Operador min.*, selecciona la menor distancia entra las dos lecturas.

Como se vio en el diseño del comportamiento anterior estas variables de error pueden tomar los siguientes valores:

$$efI = \begin{cases} +; si Setpoint > \min(medidaS11, medidaS12) \\ 0; si Setpoint = \min(medidaS11, medidaS12) \\ -; si Setpoint < \min(medidaS11, medidaS12) \end{cases}$$

Entonces la interpretación de esta variable para este controlador se resumen en:

- +: El objeto izquierdo está dentro en la zona de detección (DPWMEI=↑).
- 0: El objeto izquierdo está en el límite de la zona de detección (DPWMEI=↑).
- -: El objeto izquierdo está alejado la zona de detección (DPWMEI=0).

La figura 4.22 muestra cómo debería actuar el robot si detecta un objeto a su costado frontal izquierdo.

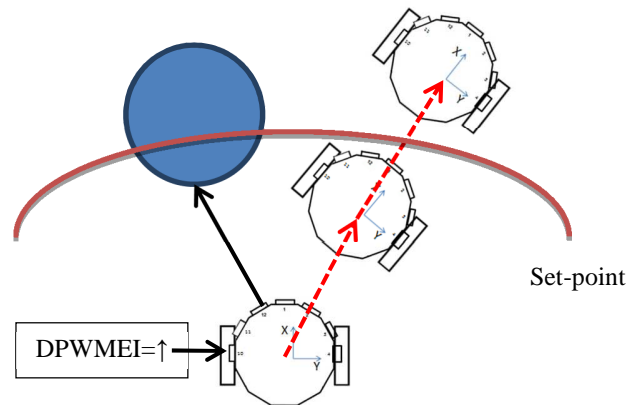


Figura 4.22. Detección obstáculo costado frontal izquierdo.

En la detección de objetos frontales derechos se realizó una asociación sensorial de S2 y S3. La detección y discriminación del objeto se realizó mediante la siguiente ecuación:

$$efD = Setpoint - \min(medidaS2, medidaS3)$$

Dónde:

- efD = Error frontal derecho
- $Setpoint$ = 75cm (referencia propuesta).
- $medidaS2$, medida actual del sensor 2.

- *medidaS3*, medida actual del sensor 3.
- *Operador min.*, selecciona la menor distancia entre las dos lecturas.

La variable del sensor frontal se puede expresar de la siguiente manera:

$$efD = \begin{cases} +; & \text{si } Setpoint > \min(medidaS2, medidaS3) \\ 0; & \text{si } Setpoint = \min(medidaS2, medidaS3) \\ -; & \text{si } Setpoint < \min(medidaS2, medidaS3) \end{cases}$$

Entonces la interpretación de esta variable para este controlador se resume en general como:

- +: El objeto derecho está dentro en la zona de detección (DPWMED=↑).
- 0: El objeto derecho está en el límite de la zona de detección (DPWMED=↑).
- -: El objeto derecho está alejado la zona de detección (DPWMED=0).

Figura 4.23 a) muestra la detección de obstáculos frontales derechos y la figura 4.23 b) muestra la detección de obstáculos frontales.

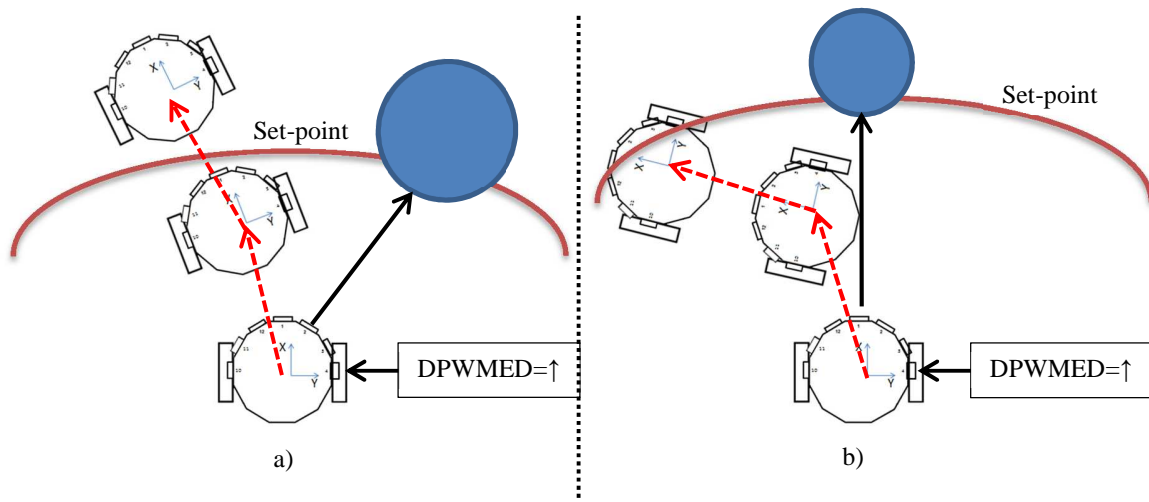


Figura 4.23. Detección de obstáculos frontales derechos. b) Detección de obstáculos frontales.

En la detección de obstáculos frontales se utilizó el sensor S1. La ecuación que establece la ecuación de los errores hacia los objetos directamente frontal es la siguiente:

$$eS1 = Setpoint - medidaS1$$

Dónde: *Setpoint* = 50cm (referencia propuesta).

- *medidaS1*, medida actual del sensor 1.

Se infiere entonces lo siguiente:

$$eS1 = \begin{cases} +; si Setpoint > medidaS1 \\ 0; si Setpoint = medidaS1 \\ -; si Setpoint < medidaS1 \end{cases}$$

El signo del resultado de la variable de error se puede interpretar como lo siguiente:

- +: El objeto frontal está dentro en la zona de detección (DPWMD=↑).
- 0: El objeto frontal está en el límite de la zona de detección (DPWMD=↑).
- -: El objeto frontal está alejado la zona de detección (DPWMD=0).

La figura 4.24 presenta el diagrama de bloque del controlador difuso, en éste se puede observar la dinámica general del comportamiento general de la evasión de obstáculos.

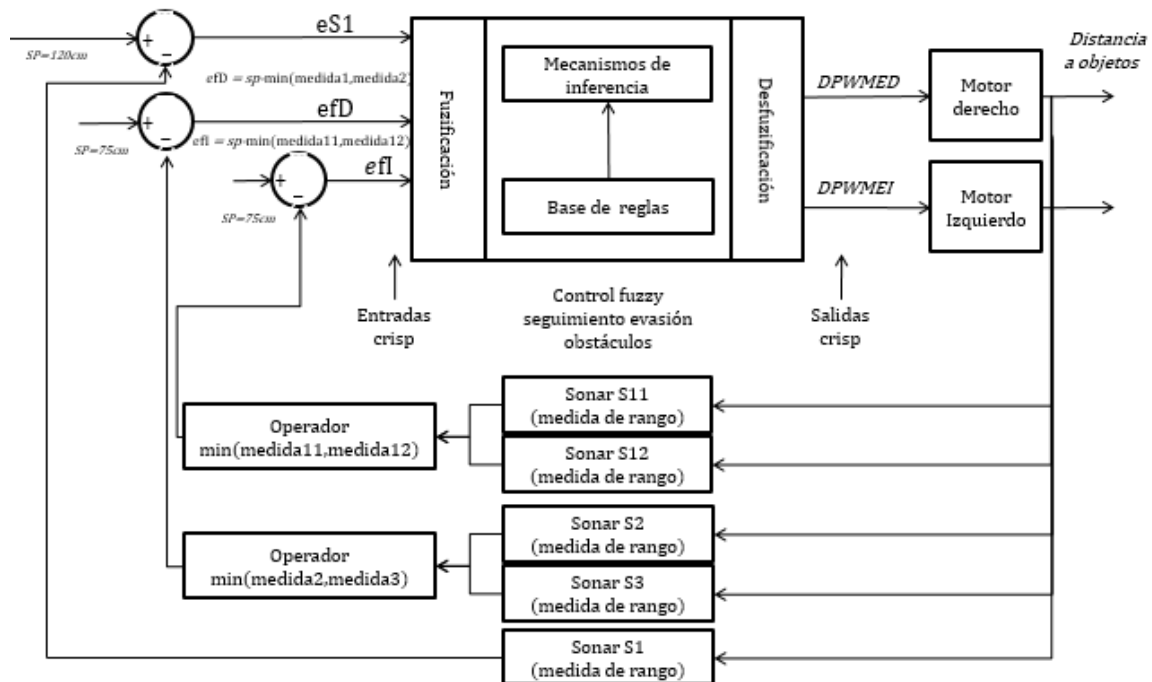


Figura 4.24. Diagrama de control *fuzzy* evasión de obstáculos.

Se observa entonces en el diagrama de control que aunque los sensores utilizados para este comportamiento son cinco, la fusión sensorial con el operador min nos reduce las variables de entrada a tres, que por ende reduce los conjuntos difusos de entrada. La estrategia del uso del operador min tuvo como objetivo simplificar el controlado para que fuese más sencillo pero que mantuviera la robustez necesaria para realizar acciones de control adecuadas.

La figura 4.25 muestra una perspectiva del robot tenemos. En esta se puede ver el proceso de sensado, cálculo de variables de entrada y variables de salida que componen el comportamiento de evasión de obstáculos

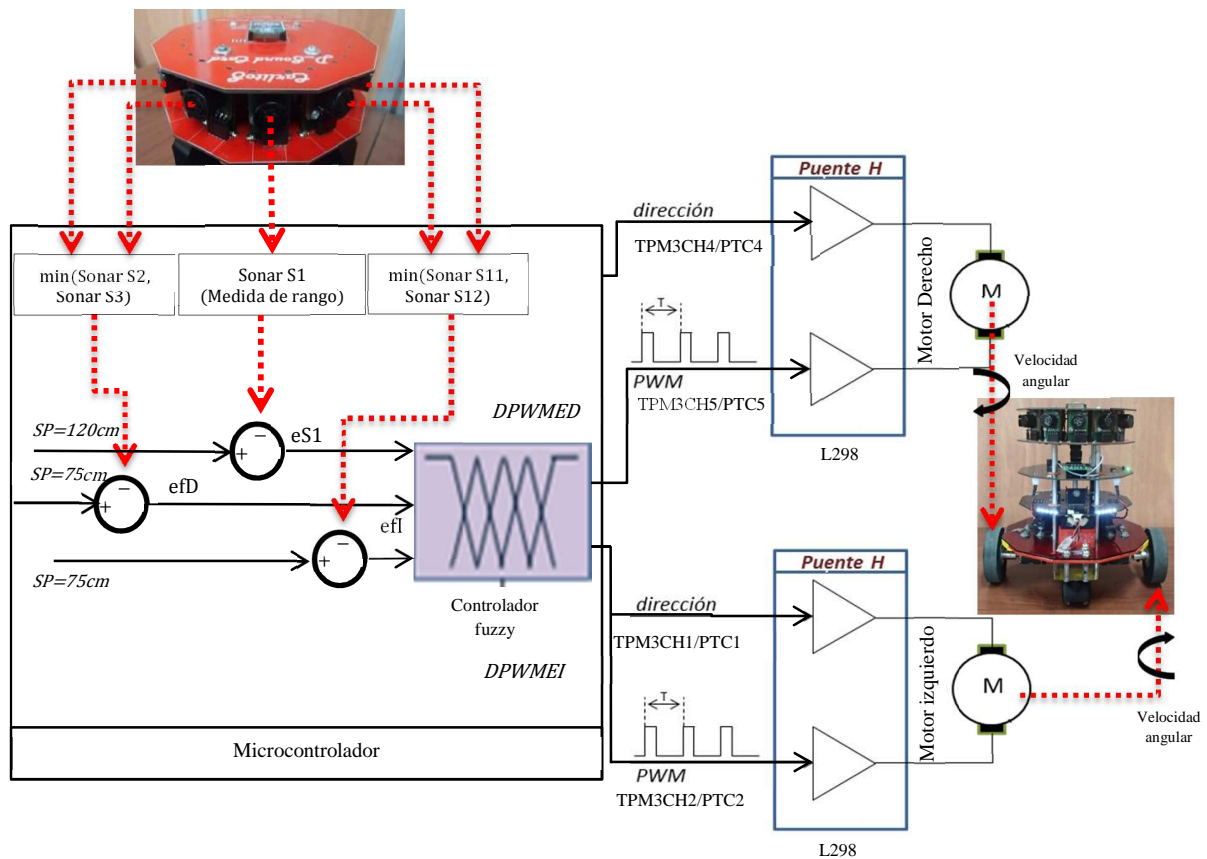


Figura 4.25. Estructura de control de evasión de obstáculos.

4.3.2. Controlador difuso

4.3.2.1. Caracterización de variables lingüísticas para el comportamiento evasión de obstáculos.

Las siguiente tablas muestras la caracterización de las variables del error frontal izquierdo, error frontal derecho, error de S1 y las salidas DPWMED e DPWMEI utilizadas para el diseño de este comportamiento.

Las tablas 14 a la presentas la caracterización de las variables lingüísticas utilizadas en el comportamiento de evasión de obstáculos.

CARACTERIZACIÓN DE LA VARIABLE LINGÜÍSTICA DEL ERROR FRONTAL IZQUIERDO	
Nombre: efl	
Tipo: entrada trapmf	Tipo: entrada trapmf
Conjunto de términos: efl_Negativo. (efl_Nt)	Conjunto de términos: efl_Positivo. (efl_Pt)
Universo de discusión: U=[-10 10]	Universo de discusión: U=[-10 10]
Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: efl_Nt = [-16 -10 -5 0]	Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: efl_Pt = [0 5 10 16]

Tabla 14. Caracterización variable de error frontal izquierdo.

CARACTERIZACIÓN DE LA VARIABLE LINGÜÍSTICA DEL ERROR FRONTAL DERECHO	
Nombre: efD	
Tipo: entrada trapmf	Tipo: entrada trapmf
Conjunto de términos: efD_Negativo. (efD_Nt)	Conjunto de términos: efD_Positivo. (efD_Pt)
Universo de discusión: U=[-10 10]	Universo de discusión: U=[-10 10]
Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: efD_Nt = [-16 -10 -5 0]	Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: efD_Pt = [0 5 10 16]

Tabla 15. Caracterización variable de error frontal derecho.

CARACTERIZACIÓN DE LA VARIABLE LINGÜÍSTICA DEL ERROR eS1	
Nombre: eS1	
Tipo: entrada trapmf	Tipo: entrada trapmf
Conjunto de términos: eS1_Negativo. (efD_Nt)	Conjunto de términos: eS1_Positivo. (efD_Pt)
Universo de discusión: U=[-10 10]	Universo de discusión: U=[-10 10]
Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: efD_Nt = [-16 -10 -5 0]	Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: efD_Pt = [0 5 10 16]

Tabla 16. Caracterización variable de error S1.

CARACTERIZACIÓN DE LA VARIABLE LINGÜÍSTICA DE LA SALIDA DELTA DE PWM MOTOR DERECHO				
Nombre: DPWMED				
Tipo: Salida trimf	Tipo: Salida trimf	Tipo: Salida trimf	Tipo: Salida trimf	Tipo: Salida trimf
Conjunto de términos: Delta de PWM Derecho Negativo Gande (DPWMED_NG)	Conjunto de términos: Delta de PWM Derecho Negativo Pequeño (DPWMED_NP)	Conjunto de términos: Delta de PWM Derecho Zero (DPWMED_Zr)	Conjunto de términos: Delta de PWM Derecho Positivo Pequeño (DPWMED_PP)	Conjunto de términos: Delta de PWM Derecho Positivo Gande (DPWMED_PG)
Universo de discusión: U=[-1100 1100]	Universo de discusión: U=[-1100 1100]	Universo de discusión: U=[-1100 1100]	Universo de discusión: U=[-1100 1100]	Universo de discusión: U=[-1100 1100]
Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: DPWMD_NG = [-1050 -1000 -950]	Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: DPWMD_NP = [-850 -900 -950]	Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: DPWMD_Zr = [-50 0 50]	Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: DPWMD_PP = [850 900 950]	Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: DPWMD_PG = [950 -1000 1050]

Tabla 17. Caracterización variable de salida de DPWMED.

Para la variable DPWMEI se caracterizó de la misma como se presenta en la tabla 17.

4.3.2.2. Tabla FAM

Al ser tres las variables de entrada para este comportamiento, la tabla FAM se comporta como estructura donde su profundidad está definida por la variable de eS1. La figura 4.26 presenta la tabla construida para este comportamiento.

		efD		eS1
		efD Nt	efD Pt	
efI	efI_Nt	DPWMED_Zr DPWMEI_Zr	DPWMED_PP	
	efI_Nt	DPWMED_PG DPWMEI_NP	DPWMED_PG	
	efI_Pt	DPWMEI_PP	DPWMED_Zr DPWMEI_Zr	
	efI_Pt	DPWMEI_PG	DPWMED_PG DPWMEI_NG	

Figura 4.26. Tabla FAM evasión de obstáculos.

En la tabla FAM se observa que en algunas de las reglas se necesita realizar acciones de control para los dos motores. Se evidencia entonces la características de los sistemas MIMO (Multiple Input, Multiple Output) descrito en la Figura 4.24.

Las reglas de este controlador difuso toman la siguiente sintaxis.

- 1. If (eS1 is eS1-Nt) and (efI is efI-Nt) and (efD is efD-Nt) then (DPWMEI is DPWMEI-Zr)(DPWMD is DPWMED-Zr)
- 2. If (eS1 is eS1-Nt) and (efI is efI-Nt) and (efD is efD-Pt) then (DPWMED is DPWMED-PP)

- 3. If (eS1 is eS1-Pt) and (efl is efl-Nt) and (efD is efD-Nt) then (DPWMEI is DPWMEI-NP)(DPWMED is DPWMED-PG)
- 4. If (eS1 is eS1-Pt) and (efl is efl-Nt) and (efD is efD-Pt) then (DPWMED is DPWMED-PG)
- 5. If (eS1 is eS1-Nt) and (efl is efl-Pt) and (efD is efD-Nt) then (DPWMEI is DPWMEI-PP)
- 6. If (eS1 is eS1-Nt) and (efl is efl-Pt) and (efD is efD-Pt) then (DPWMEI is DPWMEI-Zr)(DPWMD is DPWMED-Zr)
- 7. If (eS1 is eS1-Pt) and (efl is efl-Pt) and (efD is efD-Nt) then (DPWMEI is DPWMEI-PG)
- 8. If (eS1 is eS1-Pt) and (efl is efl-Pt) and (efD is efD-Pt) then (DPWMEI is DPWMEI-NG)(DPWMD is DPWMED-PG)

4.3.2.3. Simulación del comportamiento

La figura 4.27 presenta los conjuntos de entradas y salidas para el comportamiento de evasión de obstáculos.

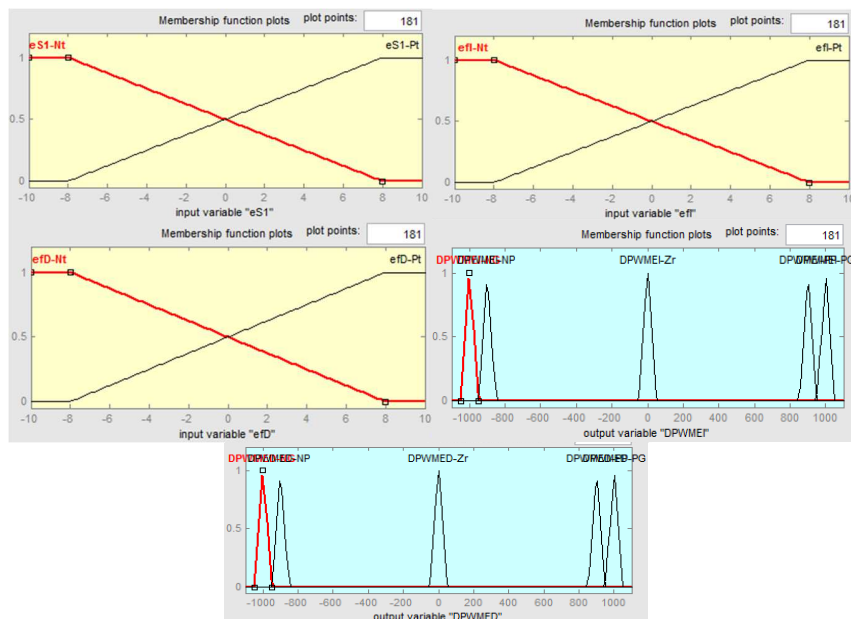


Figura 4.27. Partición de variables lingüísticas de entrada en conjuntos difusos, salida PWMD y PWMI

En la figura 4.28 se presenta las superficies de control que relacionan todas las variables de entrada respecto a las variables de salida.

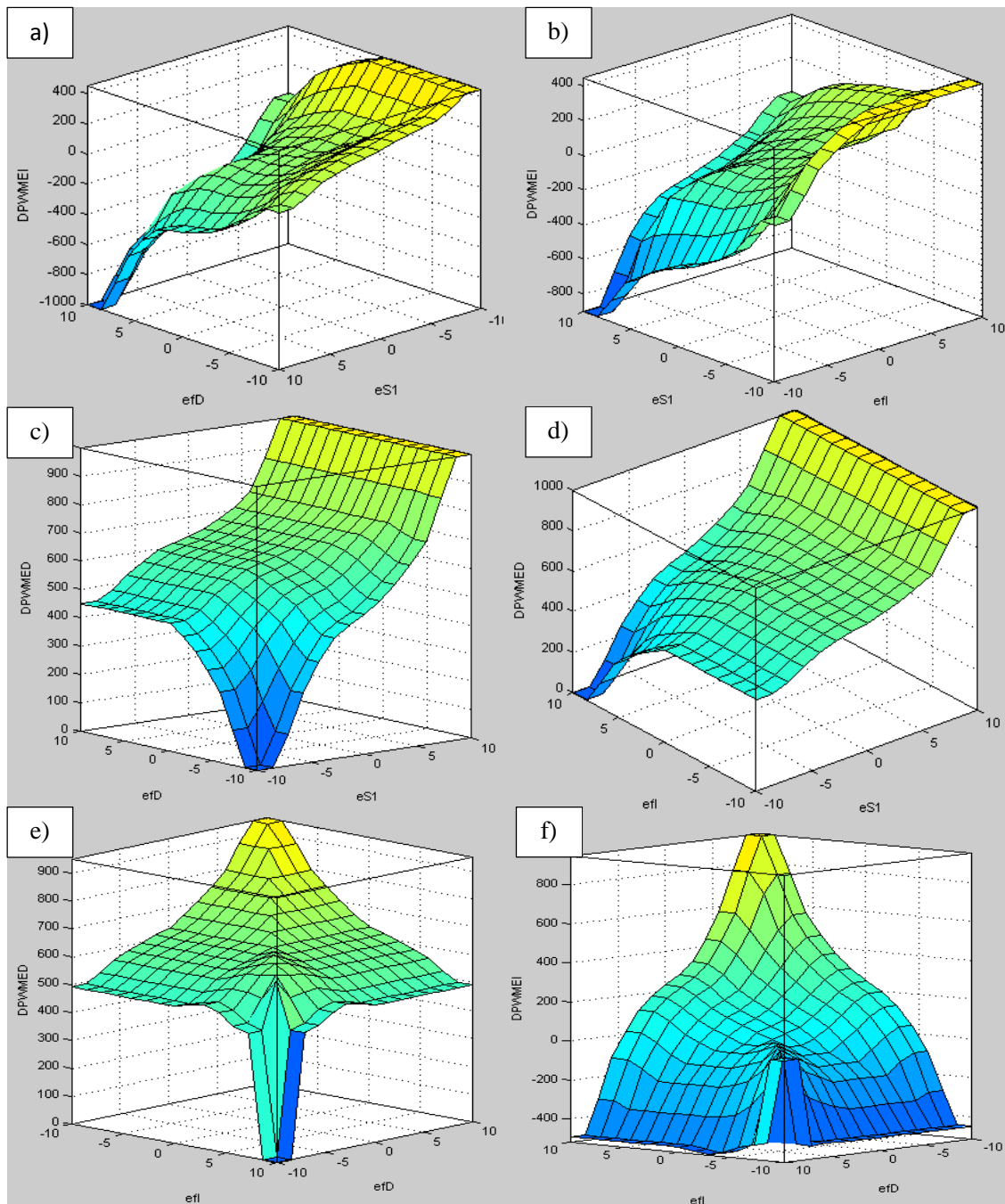


Figura 4.28. Relación de las variables de entrada eS1, efl, efD respecto a las salidas PWMD y PWMI

En el análisis de las superficies de control podemos corroborar el supuesto inicial de que las acciones de control en las tres zonas de detección tendrían una tendencia lo más lineal posible al aumento del PWM en los dos motores con el objetivo de evadir los obstáculos frontales circundantes el robot en su navegación.

En la figura 4.28 a) y la figura 4.28 b) podemos observar la relación de las variables $eS1$, efl y efD respecto al DPWMI. Estas superficies tienen la misma forma pero son diferentes en cuanto al análisis que se puede extraer de ellas. Hay que resaltar que efl es la variable que tiene prioridad en el control del motor izquierdo, por lo tanto la figura 4.28 b) es la superficie de interés para el análisis izquierdo, en la cual se constata que si hay alguna existencia de un objeto en la zona frontal izquierda (valores positivos del eje efl en la figura 4.28 b) los valores de DPWMI son incrementales para evadir el obstáculo. De la figura 4.28 a) podemos ver que la relación de detección de objetos a la derecha (ejes positivo de efD en a)) con respecto a la DPWMI es negativa debido a que en esa situación el PWMI debe ser mínimo y el PWMD máximo para evadir el obstáculo ubicado a la derecha del robot.

Para las superficies de la figura 4.28c) y la figura 4.28 d) se observan también que las superficies tienen el mismo comportamiento pero el análisis es diferente. La superficie de interés para la zona frontal derecha es la figura 4.28 c). En esta podemos ver que al igual que en la detección izquierda, sí existe un objeto en la zona de detección frontal derecha (valores positivos del eje efD en c)) los valores de DPWMD son incrementales. El límite superior de la variable de salida DPWMD en comparación con DPWMI es mayor debido a que la evasión de objetos frontales por defecto es a la izquierda (Aumento del PWM motor derecho) y en segunda instancia por que el giro de evasión de estancamiento también se decidió por defecto a la derecha el cual necesita cambios fuertes para que el robot no sufra ninguna colisión contra objetos o paredes, por estas dos razones la variable de salida DPWMD tiene un rango de mayor.

Por último, se presentan las gráficas 4.28 e) y la figura 4.28 f) donde se relacionan efl y efD . Se puede observar de las gráficas que son congruentes los supuestos planteados para el diseño en donde las detecciones para ambos lados afectarían al motor correspondiente. En la figura 4.28 e) podemos ver la relación de detección en ambos lados con respecto al DPWMD en donde priman las decisiones del error de detección a la derecha (valores positivos del eje efD en la figura 4.28 c). En esta gráfica se observa que en la mayoría de la superficie el DPWMD es positivo, esto

se debe a que en la mayoría de las situaciones las decisiones de control son incrementos hacia el motor derecho (evadir obstáculos a la derecha y girar a la derecha por defectos por razones de seguridad). Por el contrario, en la figura f) se observa que el rango de movimiento de la variable DPWMEI es más amplio incurriendo en incrementos positivos o negativos. Los incrementos positivos son interpretados como aumentos del PWMI para evadir los obstáculos presentes en la zona izquierda y la razón de los incrementos negativos se debe a que en los giros por defecto para evadir obstáculos frontales y el giro de emergencia son a la derecha, el motor izquierdo ayudara reducir sus revoluciones lo que implica cambios de DPWMI negativos para realiza las acciones de giro a la derecha con mayor rapidez evitando accidentes en estas situaciones.

4.4. DISEÑO DE COMPORTAMIENTO DE SEGUIMIENTO DE PARED IZQUIERDA

El diseño de este comportamiento es similar al de seguimiento de pared derecha. En este se espera una respuesta similar a la mostrada en la figura 4.16 pero al seguimiento de las paredes izquierdas.

Las variables de entrada para el controlador de seguimiento de paredes izquierdas son el error de medida del sensor 10 y la derivada del error de medida del sensor 10. Las condiciones de *set-point*, condiciones iniciales de navegación y de diseño para comportamiento son similares a las del seguimiento de pared derecha pero acoplada a las condiciones del seguimiento de pared derechas.

4.4.1. Diseño

Se tiene entonces para la variable del error de S10 el siguiente análisis:

$$eS10 = Setpoint - medidaS10$$

Dónde:

- *Setpoint* = 60cm (referencia propuesta).
- *medidaS10*, medida actual del sensor 10.

$$eS10 = \begin{cases} +; si \text{ Setpoint} > medida10 \\ 0; si \text{ Setpoint} = medidaS10 \\ -; si \text{ Setpoint} < medidaS10 \end{cases}$$

El signo del resultado de la variable de error se puede interpretar como lo siguiente:

- +: El robot está cercano a la pared izquierda.
- 0: El robot está manteniendo la distancia de set-point hacia la pared.
- -: El robot está alejado de la pared izquierda.

Para la derivada del error se tiene también el siguiente análisis:

$$deS10 = eS10_{(k)} - eS10_{(k-1)}$$

$$deS10 = \begin{cases} +; Derivada \text{ creciente} \\ 0; Derivada \text{ constante} \\ -; Derivada \text{ decreciente} \end{cases}$$

Al ser semejante el tratamiento de las variables de entrada utilizadas para este diseño se espera que en los análisis de navegación la dinámica del comportamiento se comporte como un espejo frente al comportamiento de seguimiento pared derecha, esto se puede evidenciar en la figura 4.29 donde se analiza el error y la derivada del error del el sensor S10

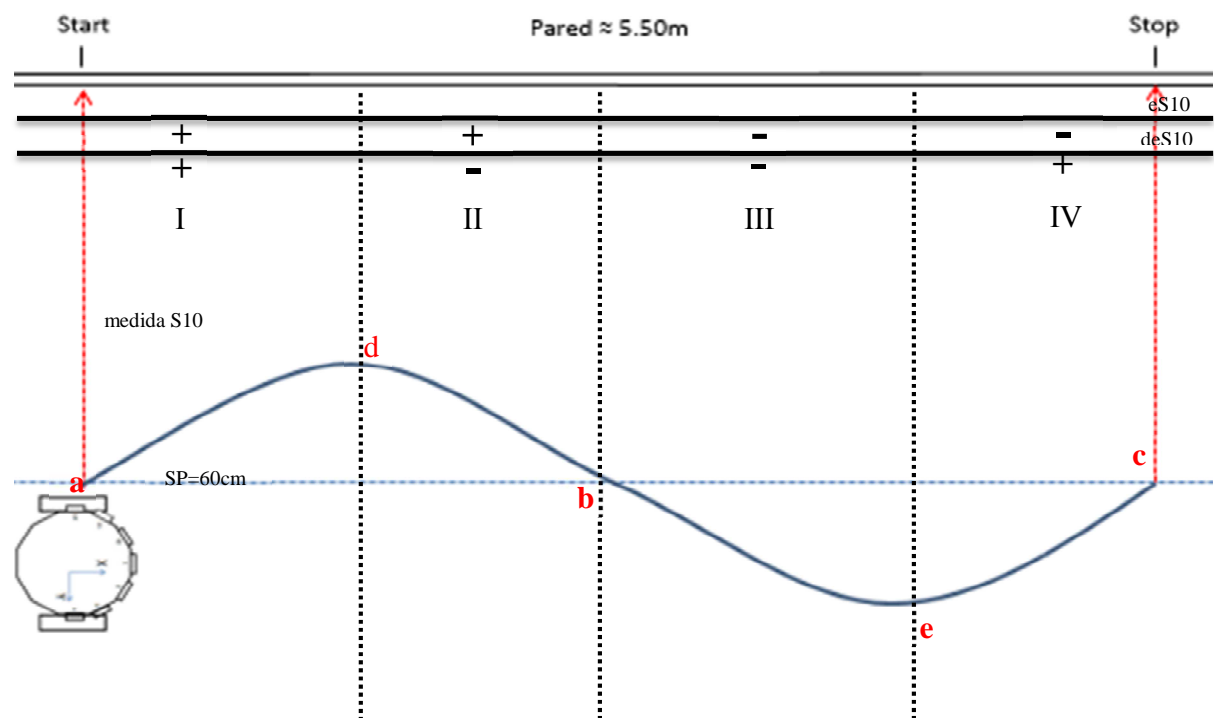


Figura 4.29. Análisis del error y la derivada del error de medida de S10.

De donde se extrajo la siguiente tabla que presenta la dinámica del comportamiento.

Regla N°	eS10	deS10($\Delta eS10$)	Acción de Control	Valor de la referencia del dibujo
1	+	+	DPWMIW= \uparrow	I
2	+	-	DPWMIW= \uparrow	II
3	-	-	DPWMIW= \downarrow	III
4	-	+	DPWMIW= \downarrow	IV
5	0	0	DPWMIW=0	a
6	0	-	DPWMIW= \downarrow	b

7	0	+	DPWMIW=↑	c
8	+	0	DPWMIW=↑	d
9	−	0	DPWMIW=↓	e

Tabla 18. Análisis del comportamiento-Seguimiento de pared izquierda.

Si se realiza una comparación entre las tablas 9 y 18 podemos inferir que respecto al diseño los comportamiento de seguimiento de pared derecha con respecto al seguimiento de pared izquierda tiene la misma dinámica y ambos son reflejo uno del otro. Tenemos entonces que las acciones de control realizadas para este comportamiento son recibidas por el motor izquierdo mediante los DPWMIW que recibe el registro encargado de controlar el canal del periférico encargado de efector final. En la figura 4.30 podemos ver el diagrama de control para este comportamiento.

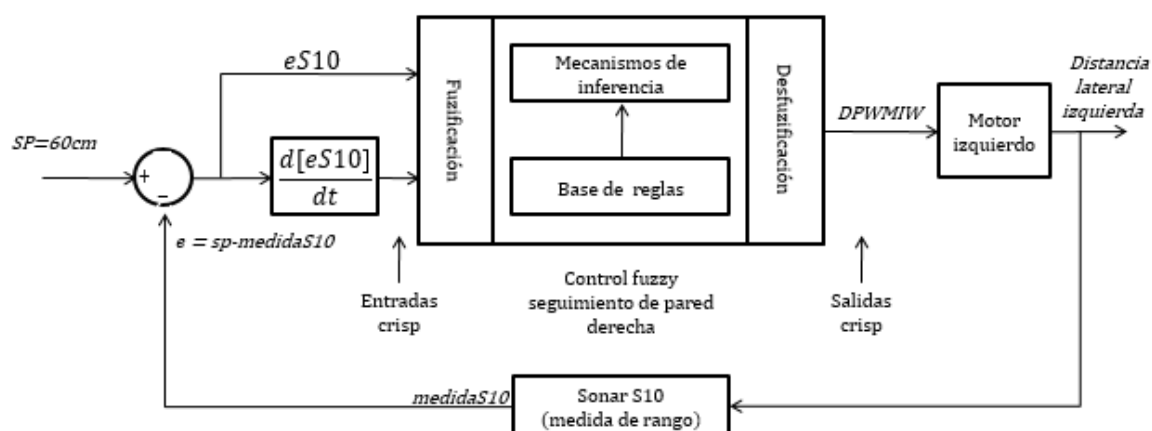


Figura 4.30. Diagrama de control *fuzzy* seguimiento de pared izquierda.

La figura 4.31 presenta una perspectiva con el robot donde se tiene lo siguiente:

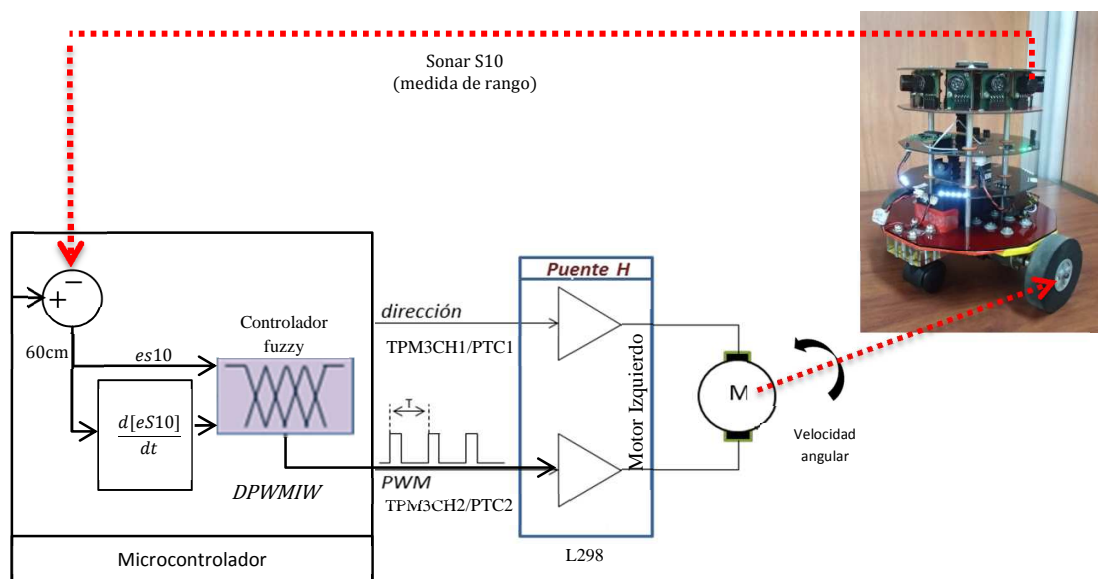


Figura 4.31. Estructura de control del seguimiento de pared izquierda.

4.4.2. Controlador difuso

4.4.2.1. Caracterización de variables lingüísticas para el comportamiento seguimiento de pared izquierda

Las tablas 19, 20 y 21 muestran la caracterización de las variables de error, la derivada del error y el delta de PWM para el motor izquierdo, dichas tablas presentan el tratamiento de las variables como conjuntos difusos.

CARACTERIZACIÓN DE LA VARIABLE LINGÜÍSTICA DEL ERROR DEL SENSOR S10		
Nombre: eS10		
Tipo: entrada trapmf	Tipo: entrada trimf	Tipo: entrada trapmf
Conjunto de términos: eS10_Negativo. (eS10_Nt)	Conjunto de términos: eS10_zero. (eS10_Zr)	Conjunto de términos: eS10_Positivo. (eS10_Pt)
Universo de discusión: U=[-20 20]	Universo de discusión: U=[-20 20]	Universo de discusión: U=[-20 20]
Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: eS4_Nt = [-30 -20 -1 0]	Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: eS4_Zr = [-1 0 5]	Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: eS4_Pt = [0 5 20 30]

Tabla 19. Caracterización variable de error seguimiento de pared izquierda.

CARACTERIZACIÓN DE LA VARIABLE LINGÜÍSTICA DE LA DERIVADA DEL ERROR DEL SENSOR deS10		
Nombre: deS10		
Tipo: entrada trapmf	Tipo: entrada trimf	Tipo: entrada trapmf
Conjunto de términos: Derivada del ErrorS10_Negativo. (deS10_Nt)	Conjunto de términos: Derivada del ErrorS10_zero (deS10_Zr)	Conjunto de términos: Derivada del ErrorS10_Positivo. (deS10_Pt)
Universo de discusión: U=[-20 20]	Universo de discusión: U=[-20 20]	Universo de discusión: U=[-20 20]
Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: deS4_Nt = [-30 -20 -1 0]	Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: deS4_Zr = [-1 0 5]	Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: deS4_Pt = [0 5 20 30]

Tabla 20. Caracterización variable derivada del error del seguimiento de pared izquierda.

CARACTERIZACIÓN DE LA VARIABLE LINGÜÍSTICA DE LA SALIDA DELTA DE PWM MOTOR IZQUIERDO				
Nombre: DPWMIW				
Tipo: Salida trimf	Tipo: Salida trimf	Tipo: Salida trimf	Tipo: Salida trimf	Tipo: Salida trimf
Conjunto de términos: Delta de PWM Derecho Negativo Gande (DPWMIW_NG)	Conjunto de términos: Delta de PWM Derecho Negativo Pequeño (DPWMIW_NP)	Conjunto de términos: Delta de PWM Derecho Zero (DPWMIW_Zr)	Conjunto de términos: Delta de PWM Derecho Positivo Pequeño (DPWMIW_PP)	Conjunto de términos: Delta de PWM Derecho Positivo Gande (DPWMIW_PG)
Universo de discusión: U=[-200 150]	Universo de discusión: U=[-200 150]	Universo de discusión: U=[-200 150]	Universo de discusión: U=[-200 150]	Universo de discusión: U=[-200 150]
Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: DPWMIW_NG = [-140 -130 -120]	Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: DPWMIW_NP = [-210 -200 -190]	Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: DPWMIW_Zr = [10 20 30]	Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: DPWMIW_PP = [100 110 120]	Reglas semánticas: Corresponde a los significados que establecen las funciones de pertenencia en las que se particiona el universo de discusión 'U' el cual se convierte en el siguiente conjunto difuso: DPWMIW_PG = [110 120 130]

Tabla 21. Caracterización variable de salida de DPWMEI.

Se observa en la tablas 19, 20 y 21 que la caracterización final realizada de estas variables varían con respecto a las variables del comportamiento espejo “Seguimiento de pared derecha”. Los ajustes de estas variables también son el resultado de la experimentación y el aporte del conocimiento del entorno por el investigador en las pruebas controladas.

Se presenta entonces la tabla FAM que comprende la asociación intuitiva de las situaciones que el robot podría encontrarse en la navegación con el comportamiento de seguimiento de pared izquierda. En la tabla 22 se encuentra la tabla FAM elaborada para este diseño.

4.4.2.2. Tabla FAM

eS10 \ deS10	deS10_Nt	deS10_Zr	deS10_Pt
eS10_Nt	DPWMIW_NP	DPWMIW_NG	DPWMIW_NP
eS10_Zr	DPWMIW_NP	DPWMIW_ZR	DPWMIW_PP
eS10_Pt	DPWMIW_PP	DPWMIW_PG	DPWMIW_PP

Tabla 22. Tabla FAM seguimiento de pared izquierda.

La sintaxis de las reglas para el mecanismo de inferencia de este comportamiento es similar a las presentadas en el primer comportamiento diseñado. En la siguiente declaración se puede ver como se realiza la construcción de la regla resaltada en verde de la tabla FAM propuesta.

- If (eS10 is eS10-Nt) and (deS10 is deS10-Nt) then (DPWMEI is DPWMIW-NP)

Teniendo claro la sintaxis de las reglas, se presentan las nueve reglas que componen este comportamiento.

- 1. If (eS10 is eS10-Nt) and (deS10 is deS10-Nt) then (DPWMIW is DPWMIW-NP)
- 2. If (eS10 is eS10-Nt) and (deS10 is deS10-Zr) then (DPWMIW is DPWMIW-NG)
- 3. If (eS10 is eS10-Nt) and (deS10 is deS10-Pt) then (DPWMIW is DPWMIW-NP)

- 4. If (eS10 is eS10-Zr) and (deS10 is deS10-Nt) then (DPWMIW is DPWMIW-NP)
- 5. If (eS10 is eS10-Zr) and (deS10 is deS10-Zr) then (DPWMIW is DPWMIW-Zr)
- 6. If (eS10 is eS10-Zr) and (deS10 is deS10-Pt) then (DPWMIW is DPWMIW-PP)
- 7. If (eS10 is eS10-Pt) and (deS10 is deS10-Nt) then (DPWMIW is DPWMIW-PP)
- 8. If (eS10 is eS10-Pt) and (deS10 is deS10-Zr) then (DPWMIW is DPWMIW-PG)
- 9. If (eS10 is eS10-Pt) and (deS10 is deS10-Pt) then (DPWMIW is DPWMIW-PP)

4.4.2.3. Simulación del comportamiento

5. De la figura 4.32 podemos ver entonces que con las particiones realizadas para las variables lingüísticas se obtuvo una superficie de control que tiende a comportarse como un plano al igual que el comportamiento de seguimiento de pared derecha, Pero en ésta se observa con mayor inclinación, lo que indica que los cambios son más fuertes.

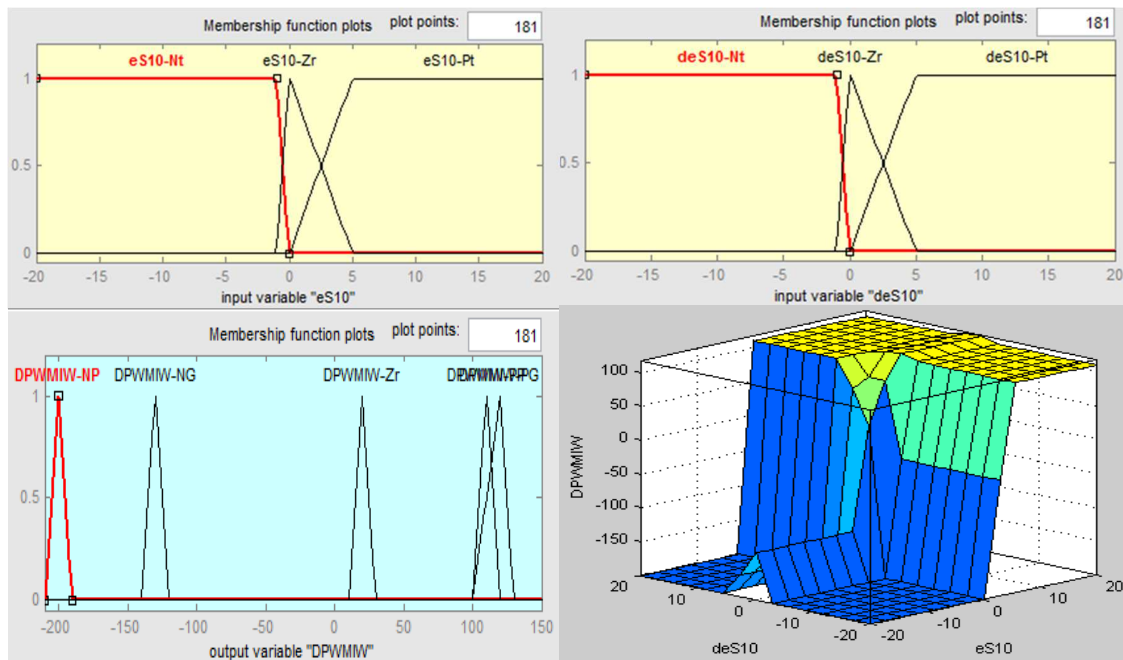


Figura 4. 32. Partición de variables lingüísticas de entrada en conjuntos difusos, salida PWMD y superficie de control para el comportamiento de seguimiento de pared izquierda.

Se observa que al igual que en los demás comportamientos se presentan asimetrías tanto en los conjuntos de entrada como en el los conjuntos de salida debido a las

ajustes realizados en la experimentación. La superficie generada en la simulación de este comportamiento presenta cambios radicales pero de menor magnitud tanto para el incremento y el decremento de los diferenciales de PWM en el motor izquierdo. La forma de esta superficie es consecuencia de las correcciones que se deben hacer en el motor izquierdo, puesto que éste interpreta los cambios de PWM más fuerte que el motor derecho, por tal razón el rango de valores para el PWI es menor que el PWMD en el seguimiento de paredes derechas para que pueda haber una concordancia en el accionamiento de los efectores.

5.1. PRUEBAS Y RESULTADOS

En esta sección se presentan los resultados obtenidos de las pruebas diseñadas. Cabe recordar que las ejecuciones fueron en escenarios controlados y estáticos delimitados por el investigador y la información se seleccionó de manera aleatoria del banco de pruebas existente para cada comportamiento.

5.1.1. Metodología

1. Luego de la correcta codificación de cada comportamiento en el microcontrolador, se procedió a realizar el chequeo individual de cada algoritmo.
2. Para las diferentes pruebas se habilitó solo un comportamiento, se seleccionó el ambiente de navegación en concordancia con el comportamiento y posteriormente se ubica el robot en la arena de navegación de manera controlada.
3. Se da la señal de inicio de navegación desde el ordenador y se inicia la navegación del robot en el entorno propuesto.
4. Se adquieren los datos de los sensores utilizado en cada comportamiento y las variables asociadas a este.
5. Luego de las experiencias recolectadas por parte del investigador, la experimentación y el análisis de la base de datos, se procede a realizar los ajustes (*tuning*) en los diferentes parámetros claves de cada algoritmo. Este proceso de ajustes conllevó un análisis intuitivo del comportamiento del robot

en las pruebas y realizar la reprogramación de los parámetros en pro de mejorar de cada comportamiento.

6. Vuelve al paso 3.

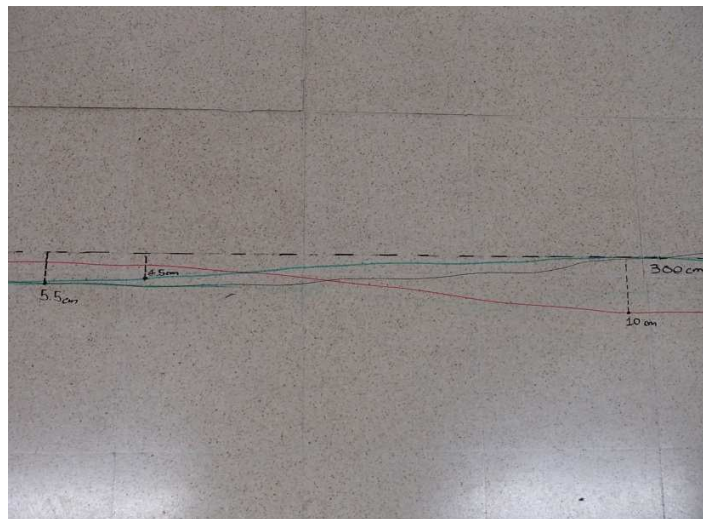
Con esta metodología se realizó un proceso de realimentación satisfactorio en el proceso de diseño de cada comportamiento.

5.1.2. Seguimiento de pared derecha

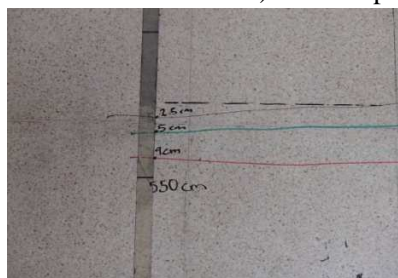
En la figura 4.33 se presentan resultados de navegaciones exitosas del robot siguiendo la pared derecha en el espacio de prueba propuesto.



a) Seguimiento de pared



b) Sobreimpulsos en la navegación



c) Error de llegada

Figura 4.33. Navegación - Seguimiento de pared derecha.

La tabla 23 presenta las mediciones tomadas experimentalmente para la prueba de seguimiento de pared derecha

Trayectoria (T)	Sobreimpulso (cm)	Error de orientación final (cm)
T_1	5.5	2.5
T_2	10	9
T_3	5.5	5

Tabla 23. Datos de las navegaciones.

Se observa una mejora notable del seguimiento de pared derecha en comparación con las pruebas iniciales en el 'CONTROL POR DECISIONES ON-OFF'. En la figura 4.33 a) se ve claramente que el robot realiza sus correcciones lo más suave posible tratando siempre de mantenerse sobre la referencia. En la figura 4.33 b) los sobreimpulsos con respecto a la referencia disminuyeron notablemente y los errores de llegada en c) tuvieron mejor exactitud.

En la figura 4.34 se presentan los datos adquiridos de una de las navegaciones realizadas en esta prueba. Los grafos describen el comportamiento de las variables medida del sensor Sa4, eS4 y deS4.

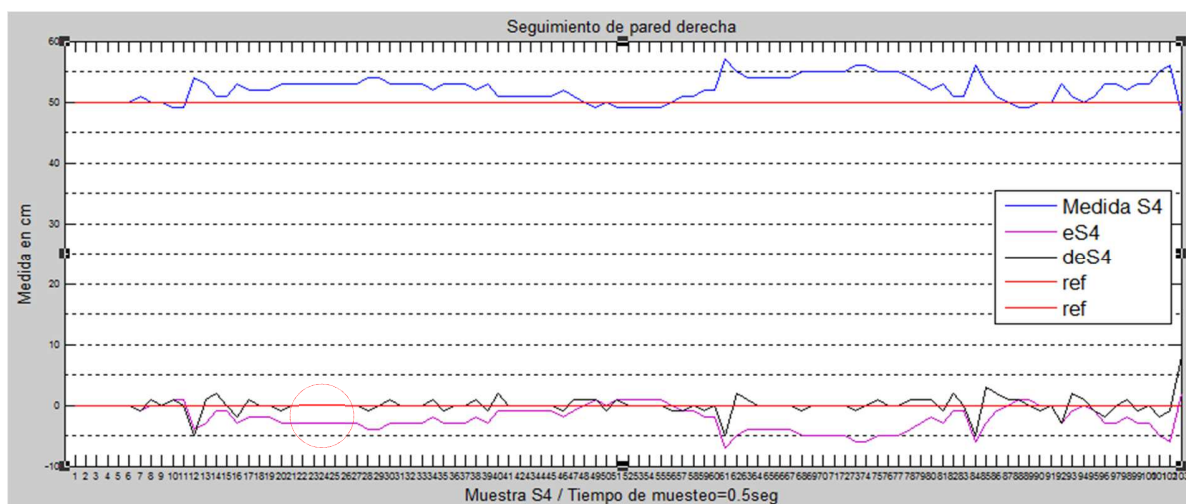
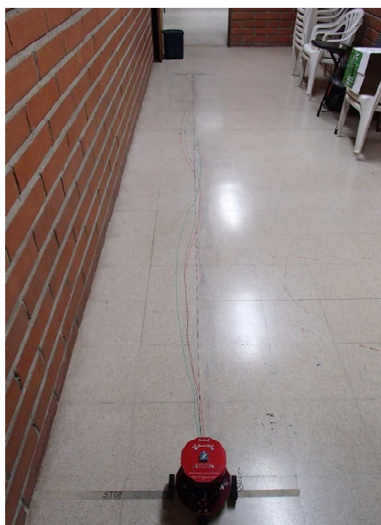


Figura 4.34. Datos suministrados por el robot.

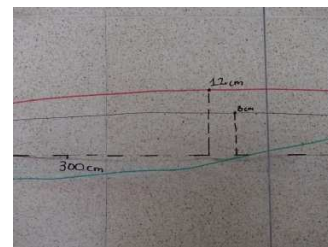
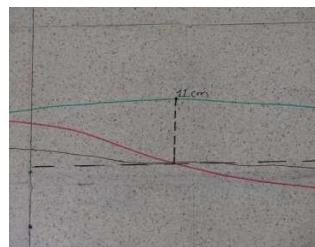
La figura muestra los datos adquiridos en el entorno de navegación propuesto en la figura 4.17. En el seguimiento del segmento lineal, en promedio los datos adquiridos oscilan entre las 100 muestras las cuales fueron guardadas como una matriz de datos que se actualiza cada 0.5 seg. La base de datos para todos los comportamientos fue adquirida en MATLAB por medio de un algoritmo de adquisición desarrollado para esta investigación. Podemos ver en la figura presentada que en general hubo cumplimiento de las acciones de control descritos en la tabla FAM construida para este comportamiento. En el círculo rojo inscrito en la figura podemos observar que la derivada es 0 (robot paralelo a la referencia) y error negativo (robot lejos de la pared) se observa entonces las correcciones correctas hacia el set-point en las próximas muestras.

5.1.3. Seguimiento de pared izquierda

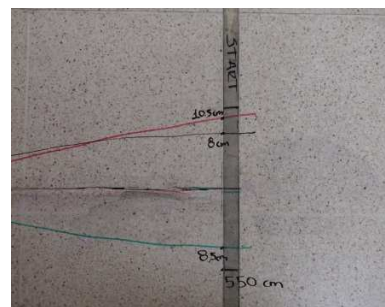
Para de pared izquierda mostrado en la figura 4.35 realizó el mismo modelo de presentación, experimentación real y datos adquiridos.



a) Seguimiento de pared



b) Sobreimpulsos en la navegación



c) Error de llegada

Figura 4.35. Navegación - Seguimiento de pared izquierda.

La tabla 24 presenta las medidas obtenidas para las pruebas de seguimiento de pared derecha.

Trayectoria (T)	Sobreimpulso (cm)	Error de orientación final (cm)
T_1	8	8
T_2	12	10.5
T_3	12	8.5

Tabla 24. Datos de las navegaciones.

Se observa que este comportamiento fue más oscilatorio que el seguimiento de pared derecha. Se puede corroborar entonces que aunque este comportamiento tuvo un diseño similar al anterior e idealmente se debería comporta como reflejo inverso del seguimiento de pared derecha, la experimentación nos indica que es un control totalmente diferente debido a que el motor a gobernar aunque tiene las mismas especificaciones, el funcionamiento en la realidad es diferente. Se observan en la figura 4.35 a) navegaciones más oscilatorias pero con tendencia hacia la zona cercana a la pared, esto se considera como una navegación exitosa ya que si el robot no pierde la pared, éste realizara acciones de control continuamente. Aunque los sobreimpulsos y el error de llegada fueron un tanto mayores en contraste con los resultados presentados en el seguimiento de pared derecha estos son aceptables.

En la figura 4.36 se puede presenciar los datos adquiridos de una las pruebas de navegación de seguimiento de pared derecha. En ella se presen las variables de entrada medida del sensor 10, eS10 y deS10

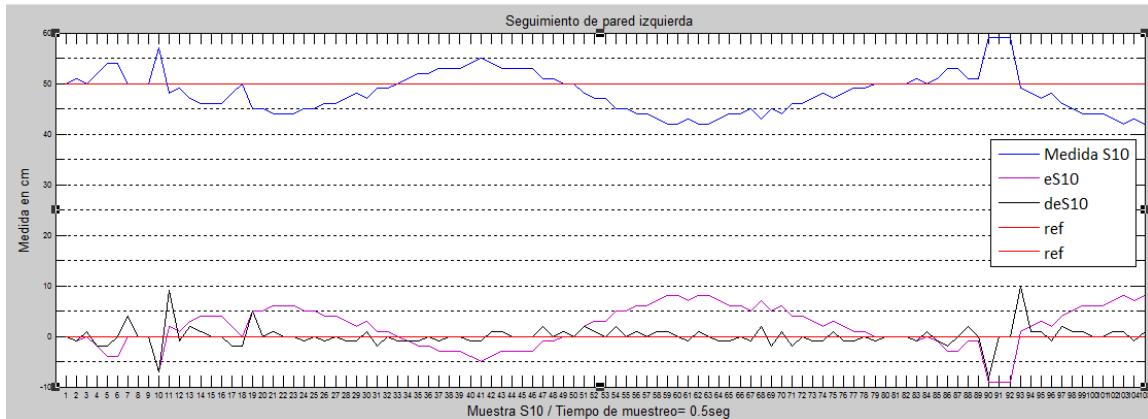


Figura 4.36. Datos suministrados por el robot.

Se observa la respuesta oscilatoria para este comportamiento en concordancia con las navegaciones descritas físicamente por el robot en la arena de experimentación. Se presencia entonces que las correcciones son correctas para las diferentes situaciones que se presentan en la navegación,

5.1.4. Evasión de obstáculos

En estas pruebas, el robot fue expuesto a escenarios controlados con objetos estáticos en donde se realizó una abstracción intuitiva a priori de la respuesta que debería tener el robot. El robot reaccionó arrojando navegaciones exitosas aproximadas al supuesto planteado inicialmente. En la gráfica 4.37 se presentan los resultados de navegación en el escenario propuesto para realizar las pruebas de evasión de obstáculos.

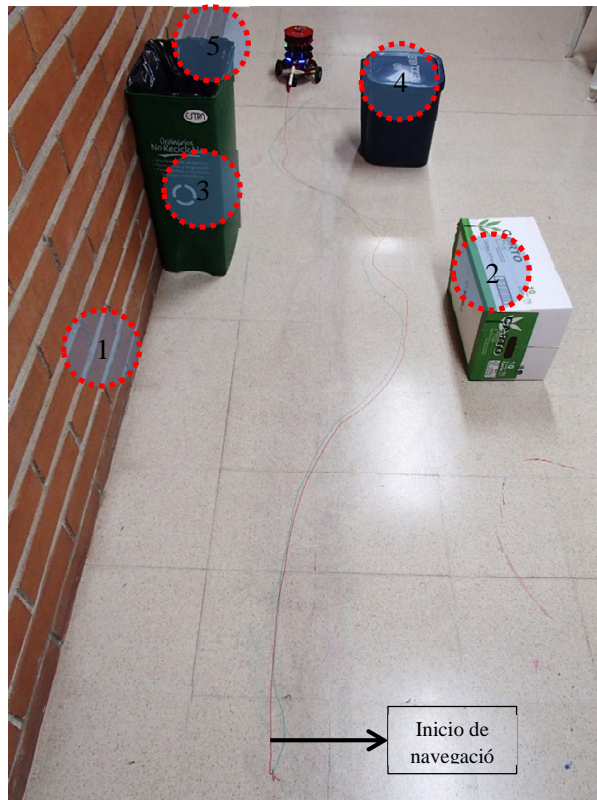


Figura 4.37. Escenario de navegación propuesto para pruebas de evasión de obstáculos.

El escenario propuesto para el chequeo de este comportamiento consta de obstáculos comunes que pudiese encontrar el robot en su navegación (canecas de basura, cajas, cajones, paredes. etc). Los cuatro obstáculos descritos en la figura 4.37 fueron posicionados por cuestiones de chequeo de tal manera que se pudiera predecir la navegación del robot. Se observa entonces que las pruebas de navegación fueron satisfactorias.

Se puede ver en la figura 4.37 las líneas que describen navegaciones aceptables evadiendo obstáculos a la derecha e izquierda. Las navegaciones describen cambios de dirección suaves cuando se detecta un obstáculo a ambos lados cumpliendo a los objetivos planteados para este comportamiento.

A continuación se presenta la figura 4.38 y 4.39 las cuales muestran la detección de los objetos a la izquierda y derecha del robot. Por otra parte la figura 4.40 presenta la detección de objetos frontales.

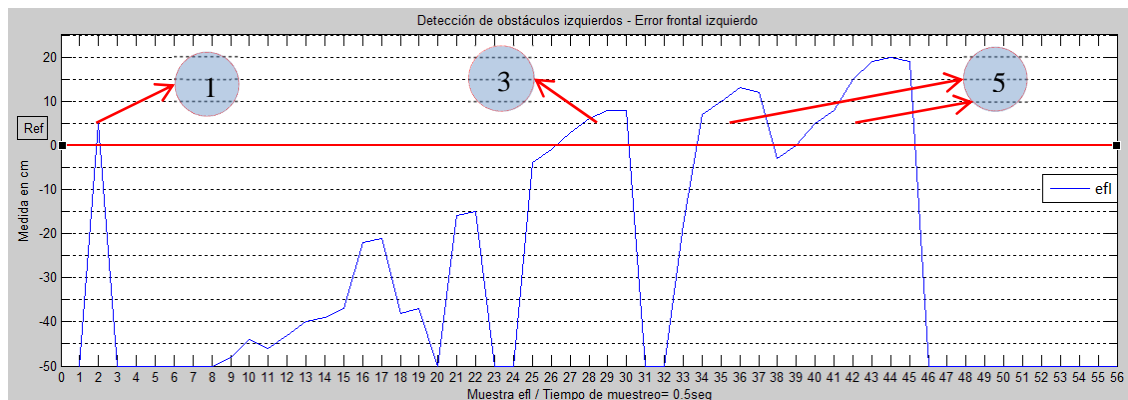


Figura 4.38. Datos de detección de objetos frontales izquierdos.

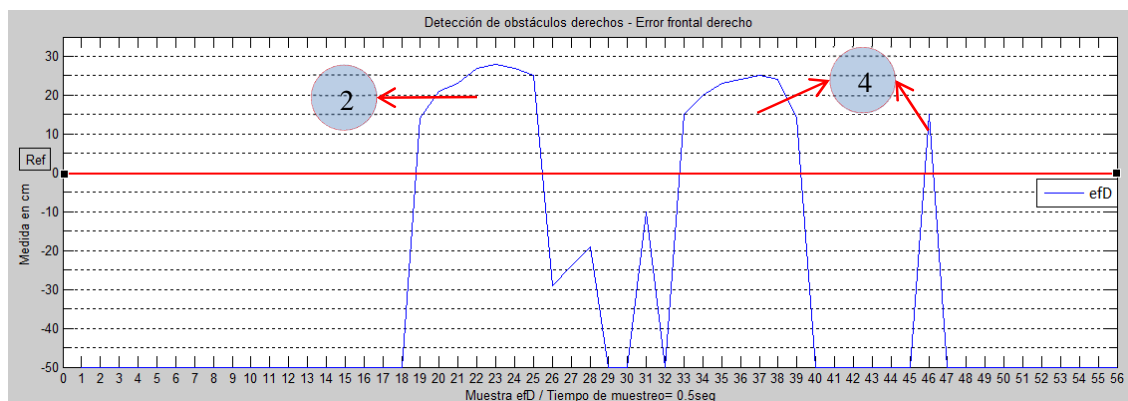


Figura 4.39. . Datos de detección de objetos frontales derechos.

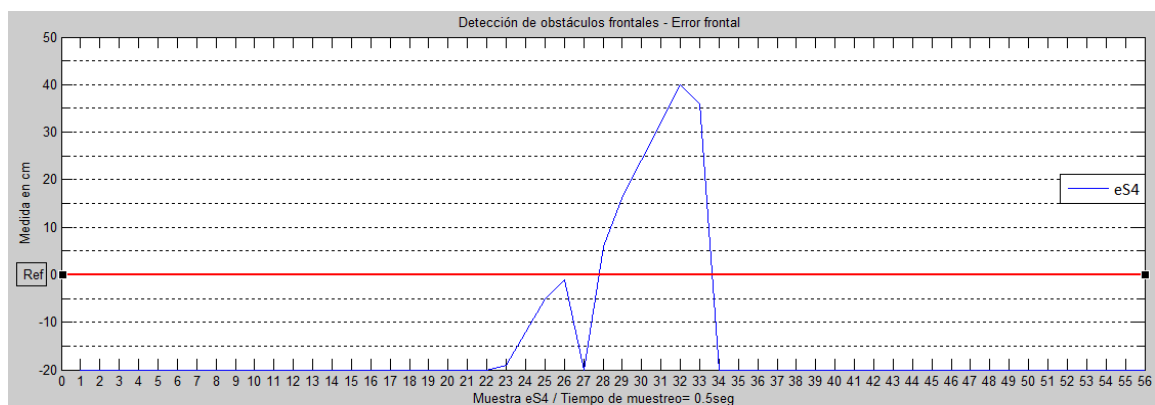


Figura 4.40. Datos de detección de objetos frontales directos.

Se infiere entonces de las gráficas presentadas que las lecturas positivas indican incidencia de objetos en las diferentes zonas monitoreadas por los diferentes sensores utilizados. En la figura 4.38 y figura 4.39 se realizó la asociación de los

picos de detección de las gráficas a los elementos que describe el entorno de prueba, podemos entonces corroborar que existe una concordancia entre las gráficas de las variables y los elementos detectados. En la figura 4.40 el objeto detectado pudo ser cualquiera de los mostrados en la arena de pruebas puesto que no existe ninguna discriminación frontal directa entre objetos

5.2. CONCLUSIONES DEL CAPÍTULO

- El conjunto de comportamiento fue diseñado satisfactoriamente cumpliendo con los lineamientos de diseño inicialmente planteados. Estos tuvieron respuestas aceptables lo cual los vuelve aptos para ser integrados en conjunto para brindar una navegación segura al robot *Carlitos*. Si bien los comportamientos ya poseen una buena respuesta todavía queda abierta la posibilidad de seguir haciendo esfuerzos en el *tuning* en pro del mejoramiento de estos.
- La experimentación exhaustiva conllevó a resultados satisfactorios generando una sólida base de conocimiento en cuanto al diseño de comportamiento de seguimiento de paredes, evasión de obstáculos y en general brindó valiosas experiencias que podrían ser útiles para abordar de manera eficiente nuevos comportamientos que se quisieran implementar en un futuro.
- Los problemas de diseño fueron transversales en todo este capítulo. Estos se presentaron en todos los diseños realizados, quizás más fuertes en unos que en otros, pero con el esfuerzo constante de entender cada vez más los problemas asociados al diseño en las extensas jornadas de experimentación, se pudieron superar estas dificultades con éxito como se puede corroborar en la sección de pruebas y resultados de este capítulo.
- La implementación de la lógica y los aportes del experto en el diseño, jugaron un papel fundamental. La lógica difusa asistió de una forma estratégica, puntual y sencilla la materialización de las ideas que el diseñador plasmó en los comportamientos. Se puede aseverar entonces

que la lógica difusa se establece una vez más como una estrategia con gran robustez y flexibilidad para el diseño de controladores basados en comportamientos para robots móviles.

Capítulo 5.

Objetivo 3: Esquema de coordinación de comportamientos

Teniendo claridad en el diseño, composición, variables de entrada/salida y dinámica de cada uno de los comportamientos, este capítulo está direccionado a presentar el esquema de coordinación de comportamientos que sirvió como mecanismo de administración de dichas conductas en la plataforma robótica *Carlitos*.

Cabe recordar entonces de una manera muy rápida y sencilla el panorama relacionado a los mecanismos de coordinación en robots móviles. Como se presentó en el estado del arte, si un autómatas tiene programado una serie de comportamientos que utilizará durante su navegación, deberá existir un mecanismo de coordinación y una fusión para que se administre de manera eficiente estas conductas para que no se generen problemas por el apoderamiento de los efectores y se cumpla de manera satisfactoria el objetivo de navegación.

Como se observó en la figura 1.18 del capítulo uno, existen dos clases de mecanismos de coordinación de comportamientos: basados en arbitramento (competitivos) donde solo se selecciona un comportamiento el cual tomará las acciones y otros basados en comandos de fusión (cooperativos) los cuales son mecanismos cooperativos que combina las diferentes recomendaciones de cada comportamiento para tomar una decisión final.

Como soporte a los diseños del capítulo anterior, se corroboró que el uso de la lógica difusa fue fundamental para incorporar todas las experiencias tanto de expertos, como experiencias en las pruebas, que realimentaron de manera satisfactoria todo el proceso de diseño de los comportamientos. Gracias a las experiencias y los resultados obtenidos se optó por diseñar un mecanismo de

coordinación de comportamientos basado en esta clase de lógica la cual hace parte de los comandos de fusión y que también se compone como estrategia cooperativa para relacionar los comportamientos diseñados.

5.1. DISEÑO DEL ESQUEMA DE COORDINACIÓN

En este esquema de coordinación se implementaron los tres comportamientos diseñados utilizando una estrategia propia que fue analizada y materializada por el investigador como aporte en paralelo a los esquemas de coordinación basados en comportamientos existentes.

Los esquemas de control independientes vistos en el capítulo anterior fueron reducidos solo por efectos visuales y para una mejor comprensión de la nueva arquitectura, de igual manera los controladores fueron implementados como se presenta en el anterior capítulo y siguen estrictamente el diagrama de bloques para cada uno de los diseños. Con esta simplificación veremos de una forma más sencilla la propuesta del nuevo sistema de control planteado para el robot *Carlitos*. Las figuras 5.1, 5.2 a) y 5.2 b) Muestran cómo se realizó la simplificación visual.

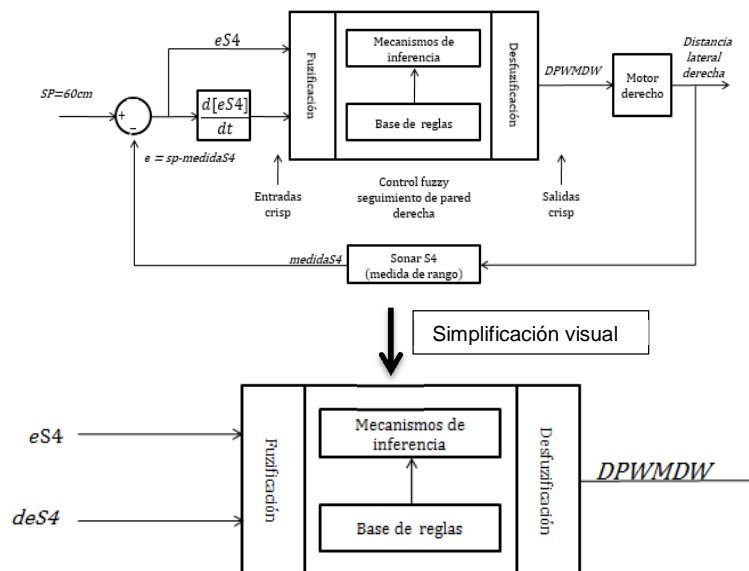


Figura 5.1. Simplificación de los sistemas de control

Entonces siguiendo el formato planteado tendremos bien la siguiente simplificación para el control de seguimiento de pared izquierda y evasión de obstáculos.

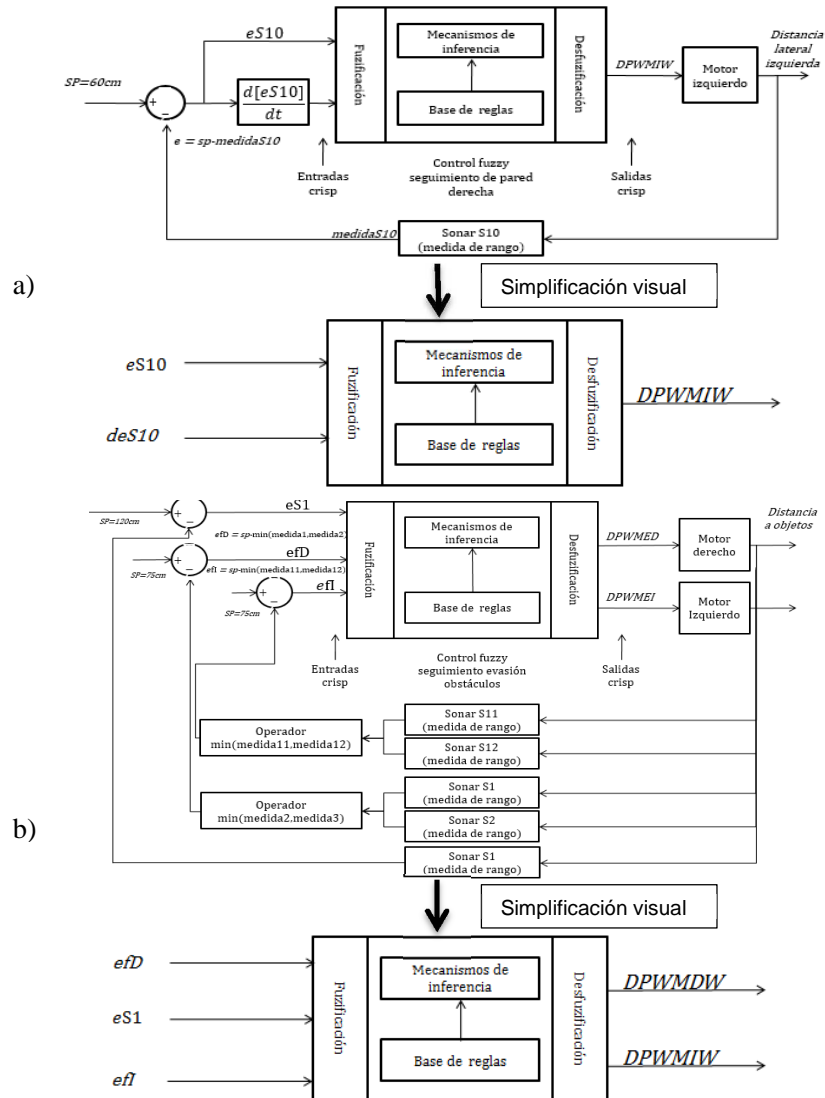


Figura 5.2. Simplificación de los diagramas a) seguimiento de pared izquierda b) evasión de obstáculos.

Esta simplificación de los diagramas de control reduce visualmente el controlado a términos de entrada y salida.

A continuación se presenta la figura 5.3, la cual enseña el diagrama de control diseñado para el robot *Carlitos*.

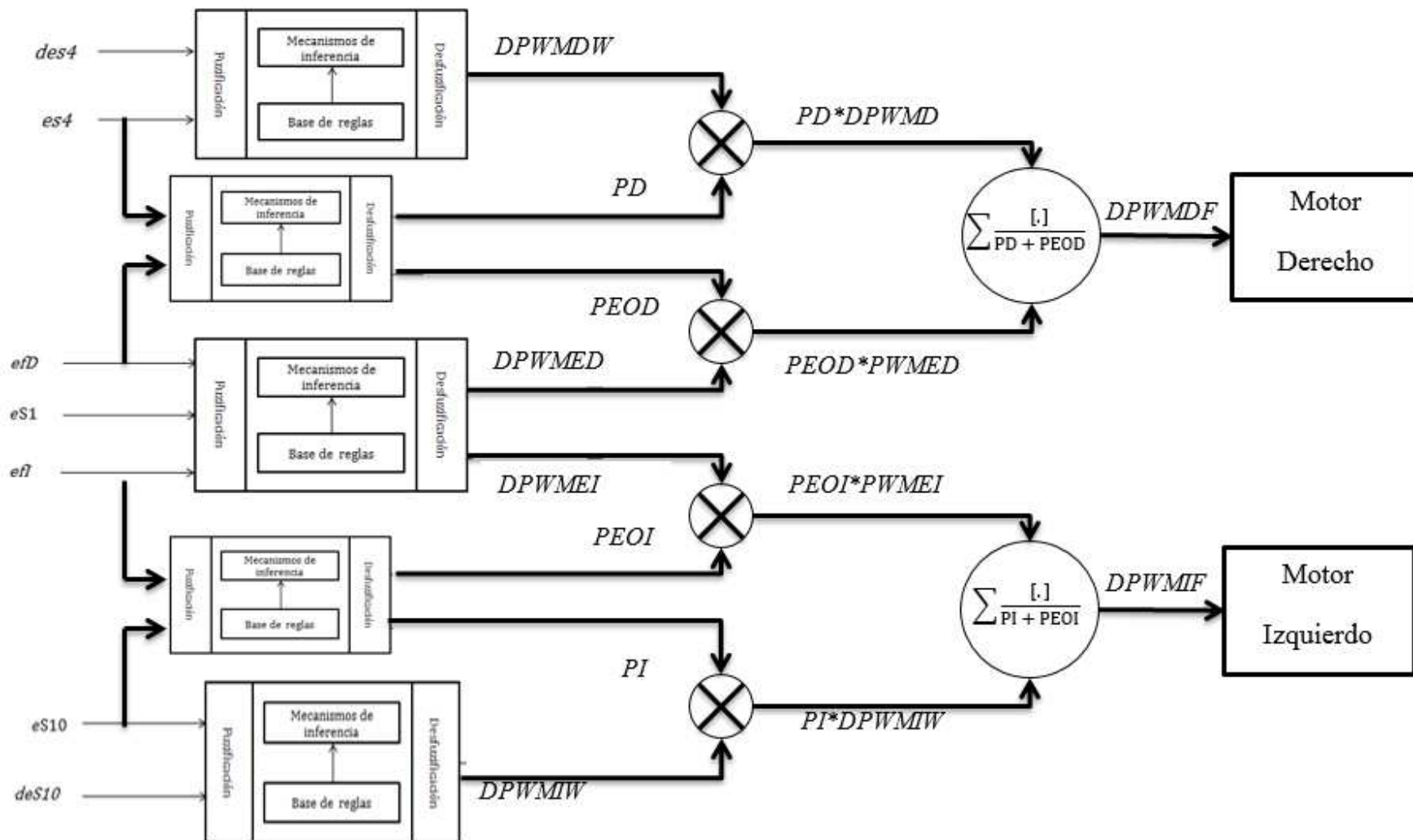


Figura 5.3. Esquema de control - Fusión por ponderación contextual dinámica difusa basada en regiones (FPCDBR).

5.2. FUSIÓN POR PONDERACIÓN CONTEXTUAL DINÁMICA BASADA EN REGIONES (FPCDBR)

Se observa en el esquema de control se adicionaron dos sistemas de inferencia difusa que se relacionan con las recomendaciones del motor derecho y el motor izquierdo por medio de factores de ponderación dinámicos difusos. Estos factores de ponderación son calculados de acuerdo a las regiones que se discriminan mediante las variables de entrada que utilizan estos sistemas de inferencia difusos adicionados.

Se tiene entonces para las variables finales de control, la fusión de las recomendaciones para cada motor con los factores de ponderación dinámicas difusas que se expresan en las siguientes ecuaciones:

$$DPW MDF = \frac{((DPW MDW) * (PD) + (DPW MED) * (PEOD))}{(PD + PEOD)} \quad (5)$$

$$DPW MIF = \frac{((DPW MIW) * (PI) + (DPW MEI) * (PEOI))}{(PI + PEOI)} \quad (6)$$

En el cálculo de los factores de ponderación dinámicos difusos realiza un aprovechamiento de las variables de error, que si se analizan detalladamente podemos presenciar que estas nos ayudan a discriminar también las regiones donde el robot navega. En la figura 5.4 se muestra el ejemplo de la discriminación del ambiente de navegación en regiones para la fusión de los comportamientos de seguimiento de pared derecha y evasión de obstáculos.

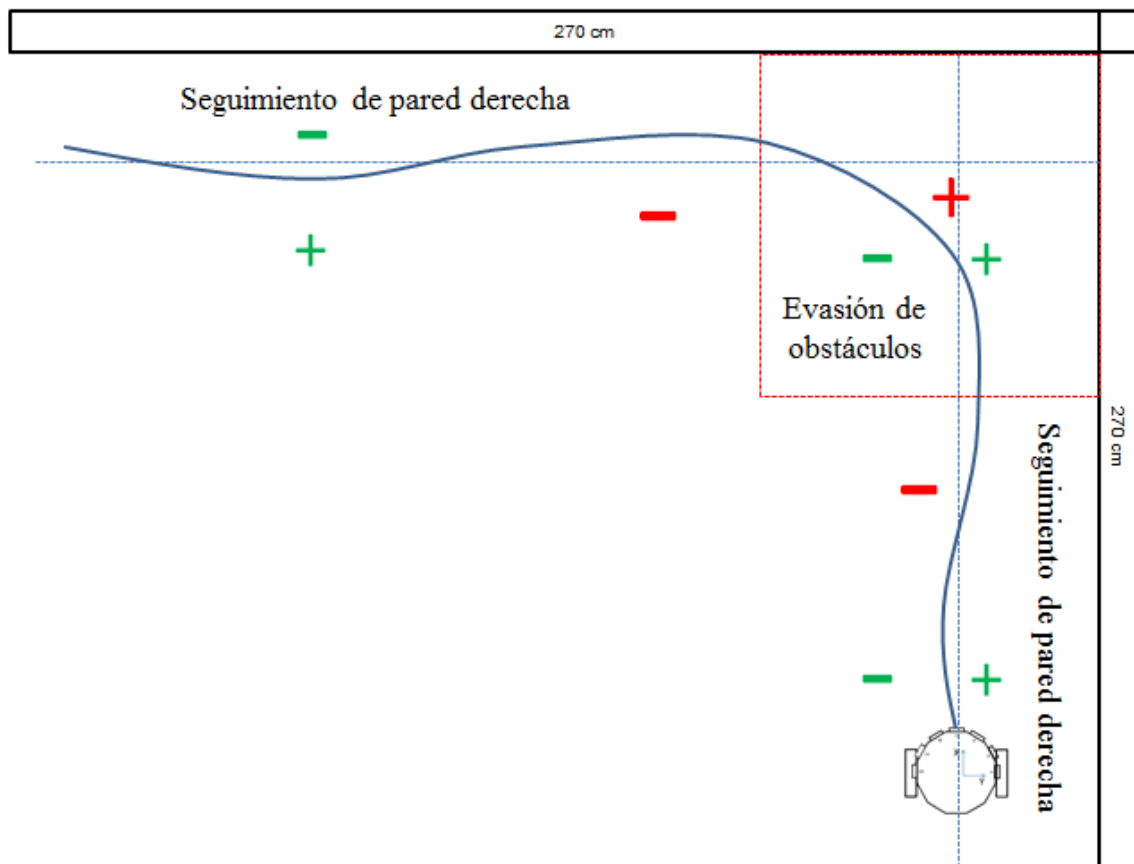


Figura 5.4. Discriminación de entorno de navegación en regiones.

Podemos ver entonces que con esta discriminación del espacio en regiones, podemos relacionar las variables de entrada del error de ambos comportamientos y dar a las recomendaciones de salida una prioridad colaborativa por medio de una ponderación dinámica difusa basada en regiones por las que el robot transita. Al igual que para el seguimiento de pared izquierda, se realizó este análisis y las simulaciones se presentan más adelante.

A continuación se presenta el sistema difuso que realiza el cálculo de los factores de ponderación dinámicos difusos que serán relacionados con las recomendaciones de los comportamientos para realizar la fusión.

5.3. DISEÑO DEL SISTEMA DIFUSO DE PONDERACIÓN

La figura 5.5 presenta la simulación del sistema de ponderación propuesto,

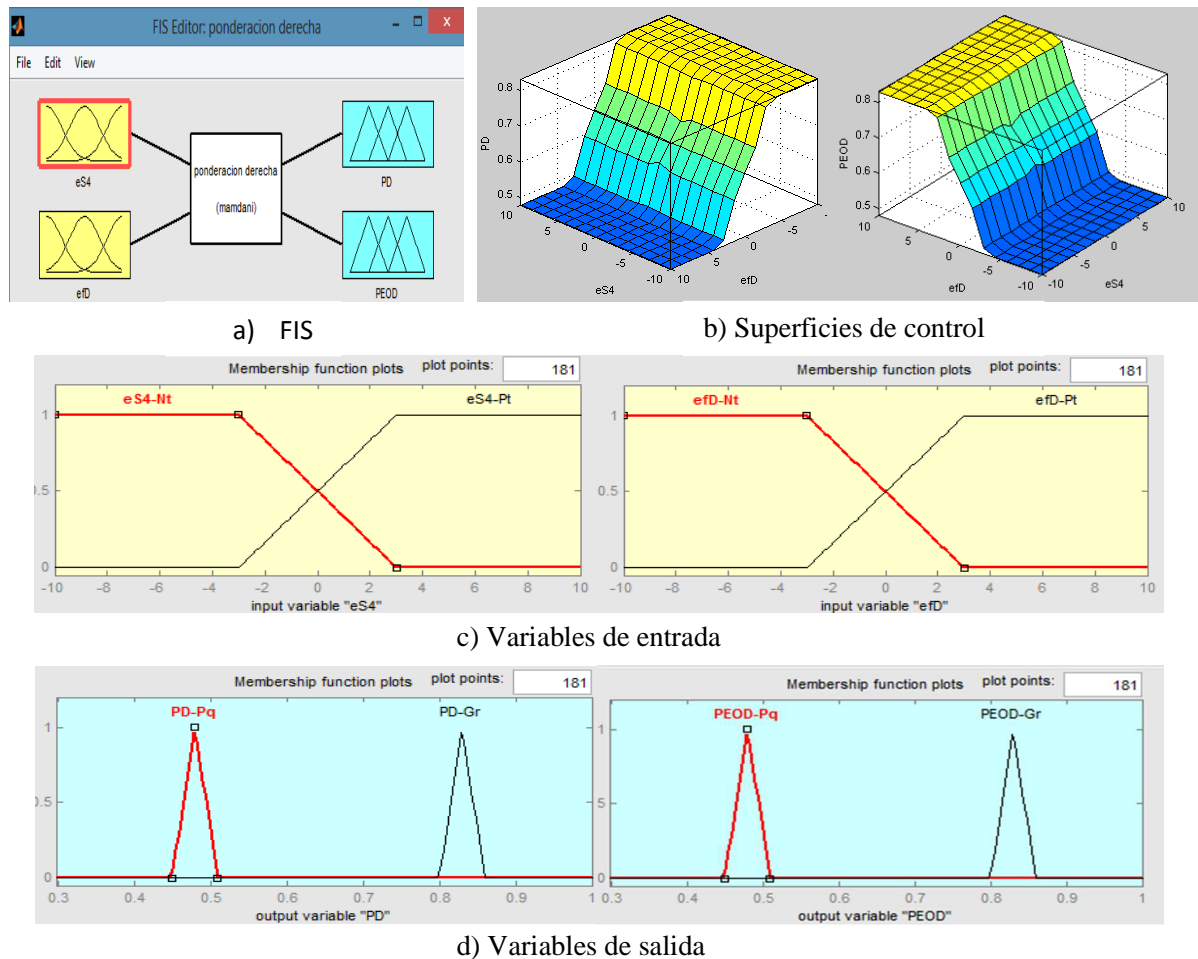


Figura 5.5. Simulación de sistema difuso para los cálculos de los factores de ponderación para las recomendaciones del motor derecho.

Se observa que superficie de control para los factores de ponderación tiende a tener una relación lineal, lo que implica que se efectúa una relación lineal de fusión de las recomendaciones de los comportamientos.

5.3.1. Tabla FAM

La tabla 25 presenta Tabla FAM asociada al contexto para el cálculo de los factores de ponderación difusos relacionado al seguimiento de pared derecha y evacion de obtaculos ubicados que se encuentren ubicados en ls zona frontal derecha.

eS4 \ efD	efD_Nt	efD_Pt
eS4_Nt	PD_Gr PEOD_Pq	PD_Pq PEOD_Gr
eS4_Pt	PD_Gr PEOD_Pq	PD_Pq PEOD_Gr

Tabla 25. Tabla FAM asociada al contexto para el cálculo de los factores de ponderación difusos.

5.3.2. Reglas de inferencia

- 1. If (eS4 is eS4-Nt) and (efD is efD-Nt) then (PD is PD-Gr)(PEOD is PEOD-Pq)
- 2. If (eS4 is eS4-Nt) and (efD is efD-Pt) then (PD is PD-Pq)(PEOD is PEOD-Gr)
- 3. If (eS4 is eS4-Pt) and (efD is efD-Nt) then (PD is PD-Gr)(PEOD is PEOD-Pq)
- 4. If (eS4 is eS4-Pt) and (efD is efD-Pt) then (PD is PD-Pq)(PEOD is PEOD-Gr)

Los resultados de la implementación de este esquema de coordinación serán presentados en el siguiente capítulo se valida por medio de una confrontación entre este diseño y una esquema de coordinación convencional utilizado en robots móviles.

5.4. CONCLUSIONES DEL CAPÍTULO

- Se observa que el diagrama de control es robusto en el cual las decisiones de los comportamientos son alimentadas por las condiciones del entorno. Esa alimentación contextual hace que el sistema este informado del entorno constante y se adapte rápidamente a él.
- La estrategia del uso de factores de ponderación dinámicos difusos mejora las decisiones que van a los motores ajustándolas en todo momento a las

- condiciones actuales del entorno, lo cual es fundamental para la navegación.
- Vemos que el aprovechamiento de las variables utilizadas para los controladores fueron reanalizadas dando como resultado una estrategia de discriminación del entorno de navegación, dicha discriminación en regiones contribuyo en pro hacia el mejoramiento de la navegación y alimentación esencial al sistema para ayudar a tomar las decisiones en los motores.

Capítulo 6.

Objetivo 4: Evaluación de esquema de coordinación de comportamientos. (Pruebas y resultados)

Para evaluar los desempeños del esquema de control diseñado en el capítulo anterior, se realizó una confrontación contra otra de las arquitecturas encontrada en la literatura. El esquema de confrontación seleccionado fue la arquitectura de subsunción o alta prioridad presentada y desarrollada por Brooks en (Brooks, 1986) la cual fue analizada y discutida anteriormente.

La subsunción o alta prioridad recomienda al sistema robótico realizar las acciones de control del comportamiento activo con mayor prioridad, así subsumiendo los demás comportamientos. En la figura 6.1 se puede ver la dinámica de tal arquitectura.

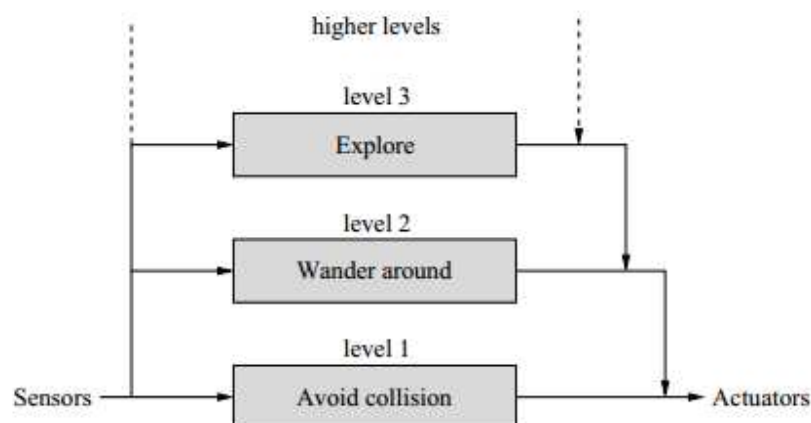


Figura 6.1. Subsunción(Paolo Pirjanian, 1999).

En donde el comportamiento con más alto nivel puede anular las decisiones hacia los actuadores de los comportamientos de más bajo nivel.

6.1. ESQUEMA DE SUBSUNCIÓN PARA EL ROBOT *Carlitos*

Llevando los comportamientos diseñados para el robot Carlitos a esta arquitectura, se tiene la siguiente relación de niveles. En la figura 6.2 se presenta la arquitectura de subsunción diseñada para el robot *CARLITOS*, la cual se puede ver de una manera más explícita en la figura 6.3

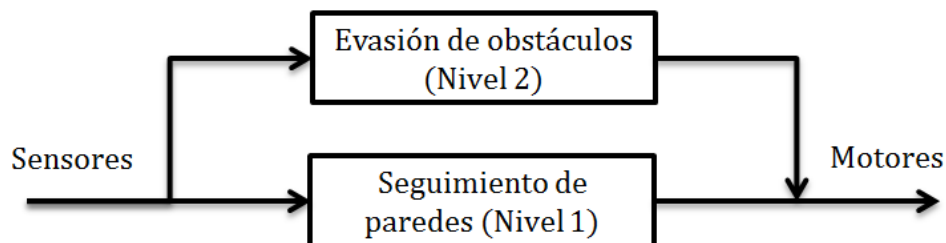


Figura 6.2. Arquitectura de Subsunción para el robot Carlitos.

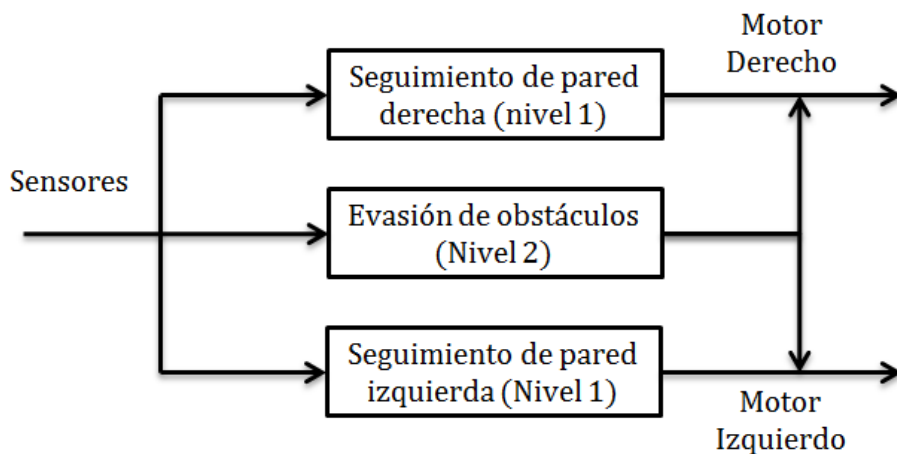


Figura 6.3. Arquitectura de Subsunción para el robot Carlitos de manera explícita.

Para la arquitectura se tiene que la evasión de obstáculos tiene mayor prioridad que los seguimientos de paredes, los cuales se encuentran en un mismo nivel con respecto a la evasión de obstáculos.

Como se apreció en las figuras 4.19, 4.25 y 4.31 los seguimientos de pared controlan el motor correspondiente a su lado y la evasión de obstáculos tiene la capacidad de alterar los dos motores, entonces para los tramos de navegación ya sea seguimiento de pared derecha y evasión de obstáculo o seguimiento de pared izquierda y evasión de obstáculos, el comportamiento de evasión tiene prioridad y puede subsumir las decisiones de los controladores de seguimiento de paredes al encontrarse algún objeto frontal, frontal derecho o frontal izquierdo en el trayecto de navegación.

6.1.2. Pruebas experimentales y resultados: esquema de subsunción

Para estas pruebas se delimitó una arena de navegación donde se pudieran presenciar la activación de los comportamientos mediante el esquema de subsunción. La figura 6.4 a) muestra el escenario de navegación de activación de los comportamientos seguimiento de pared derecha-evasión de obstáculos y en La figura 6.4 b) el escenario de navegación de activación de los comportamientos seguimiento de pared izquierda-evasión de obstáculos.

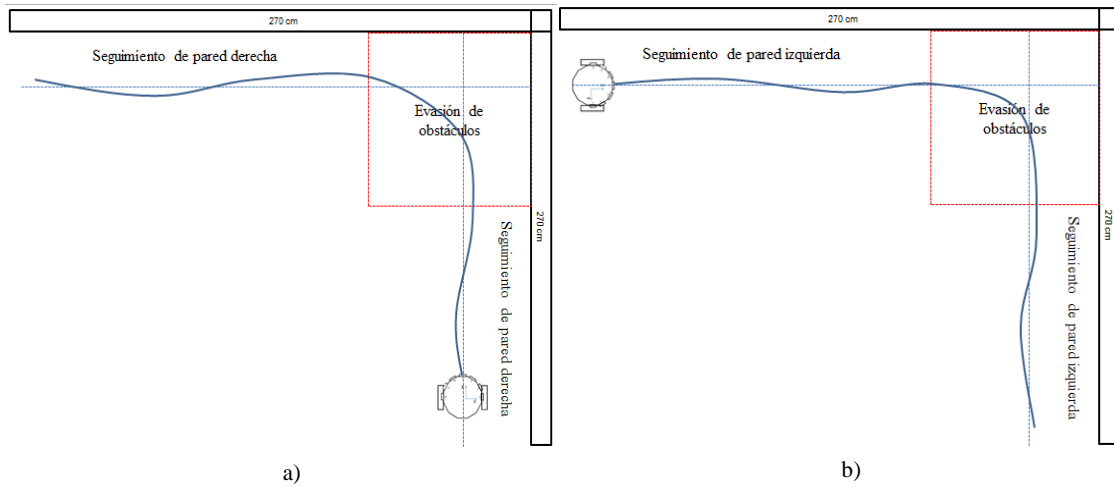


Figura 6.4.Escenario de navegación.

Se observa en la figura anterior las zonas donde se deberían activar los comportamientos según el esquema de subsunción. Para ambos gráficos la zona delimitada con rojo indica la zona donde la evasión de obstáculo tiene mayor prioridad y las zonas donde debería estar activado los seguimientos de pared en concordancia al seguimiento para el cual fue diseñado. La dinámica de estas pruebas es ver la activación del seguimiento inicial, luego la evasión de obstáculos y el retorno al seguimiento de pared. Para cada una de las pruebas se realizaron tres navegaciones las cuales se presentan a continuación.

6.1.2.1. Seguimiento de pared derecha y evasión de obstáculos

En la figura 6.5 se presenta los recorridos de navegación en donde se describen de manera experimental la navegación del robot.

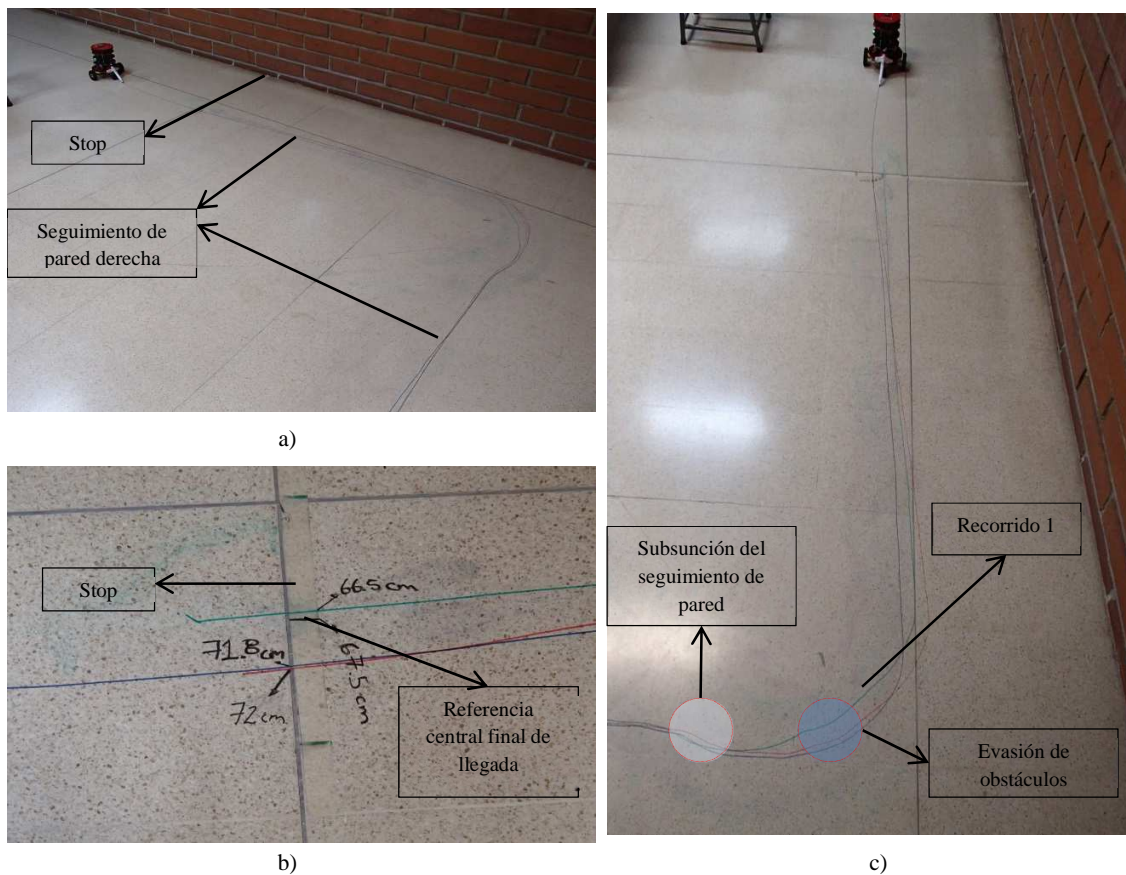


Figura 6.5. a) Seguimiento de pared derecha-evasión de obstáculos, b) Error de llegada, c) Evasión de esquina derecha.

La navegación fue satisfactoria cumpliéndose el objetivo de seguir la pared derecha y evadir obstáculos. Para estas pruebas al igual que las del seguimiento de pared izquierda, cuando el robot se aproxima hacia la pared frontal, en ese momento, ésta se comporta como un objeto frontal y se inicia el proceso de evasión de obstáculo.

Cuando el robot a evadido gran parte el obstáculo frontal, la orientación natural del robot con respecto a la pared derecha o izquierda hace que éste vea la pared como si fuese un obstáculo frontal derecho u izquierdo, lo cual es condición suficiente para así culminar la evasión total de la esquina. Posteriormente cuando el robot evade la esquina, éste retoma el seguimiento de pared derecha y termina la navegación.

Para el recorrido 1 de la gráfica 6.5 c), la figura 6.6 presenta la adquisición de datos de navegación del sensor S4, eS4 y deS4.

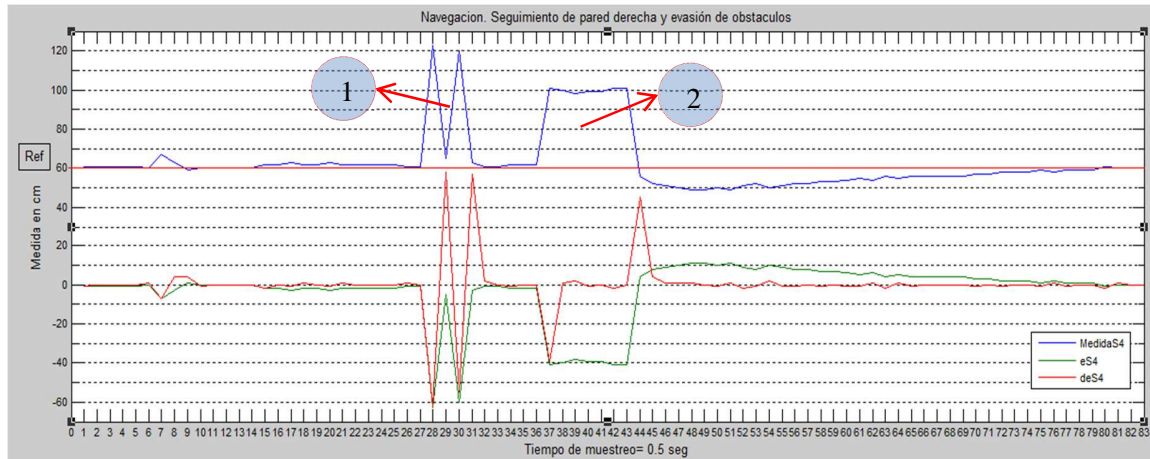


Figura 6.6. Adquisición de datos - Seguimiento de pared derecha y evasión de obstáculos.

Se observa que hay concordancia entre los datos y las imágenes de navegación presentadas anteriormente. En la etiqueta “1” de la figura 6.6 se observan reflexiones especulares contra la pared derecha, estas son mínimas y no alteran el recorrido ni la orientación del robot lo que indican robustez del controlador ante eventos de incertidumbre aleatorios.

En la etiqueta número “2” se presencia la detección del vértice de la esquina derecha. En la detección del vértice derecho se pensaría en ese momento que el robot entraría hacia la esquina para corregir el error negativo con respecto al *set-point* del seguimiento de pared derecha en esos instantes de muestreo, pero este evento no ocurre debido a que en esos instantes de navegación la evasión subsume el seguimiento de pared derecha y se efectúa la evasión sin contratiempos.

En las figuras 6.7 y 6.8 se observa la adquisición de datos relacionados al error de medida del sensor 1 y el error frontal derecho.

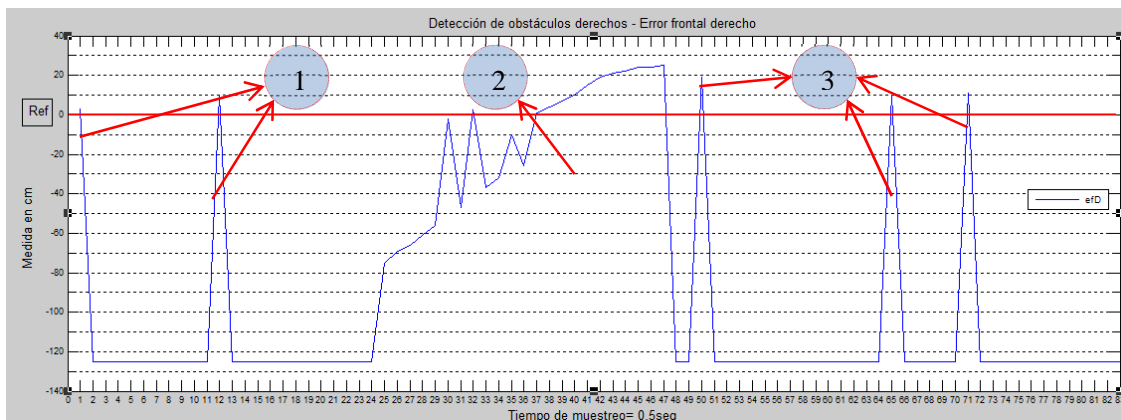


Figura 6.7. Error frontal derecho.

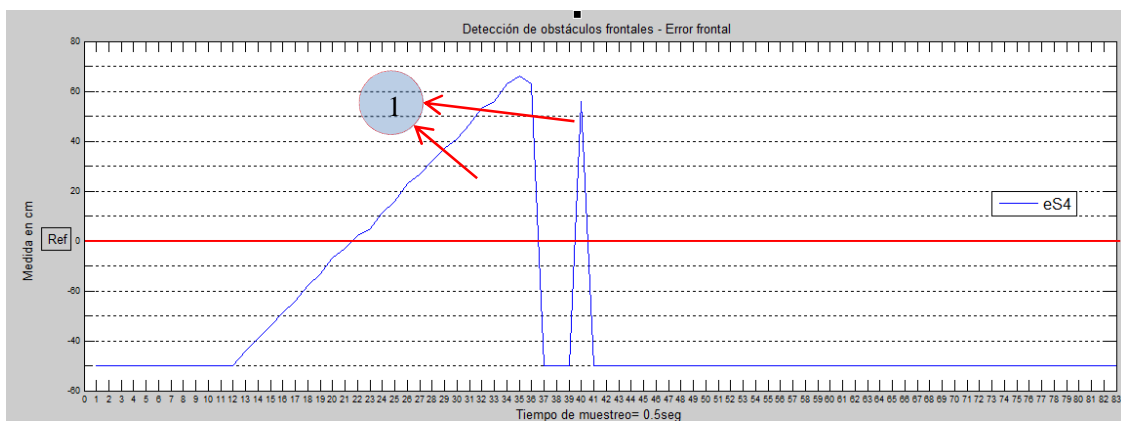


Figura 6.8. Error frontal.

La figura 6.7 presenta en 1 y 3 avistamiento de la pared derecha como un objeto que cruza la zona límite de detección, sin embargo esta influencia de la pared es mínima para que se efectúen correcciones por parte del comportamiento de evasión.

En 3 se observa una detección creciente que incide fuertemente en la zona delimitada para la evasión de objetos. Esta detección se asocia a los instantes de acercamiento del robot hacia el vértice el cual es detectado por los sensores ubicados al costado frontal derecho. En concordancia las dos figuras anteriores cuando el robot percibe la pared frontal cercana y en la zona frontal derecha hay incidencia de la pared derecha (robot muy cercano al vértice derecho) se dan errores positivos con respecto a la referencia, lo cual es una condición necesaria

para darle prioridad la evasión de obstáculos y subsumir el seguimiento de pared derecha.

6.1.2.2. Seguimiento de pared izquierda y evasión de obstáculos

Para estas pruebas se aplicó la misma metodología anterior, donde el recorrido de navegación es contrario al anterior y en el cual el robot utilizó en este caso el sensor lateral izquierdo y los encargados de detectar objeto en la zona frontal izquierda. La figura 6.9 muestra los recorridos de navegación descritos por el robot en las pruebas de seguimiento de pared izquierda y evasión de obstáculos

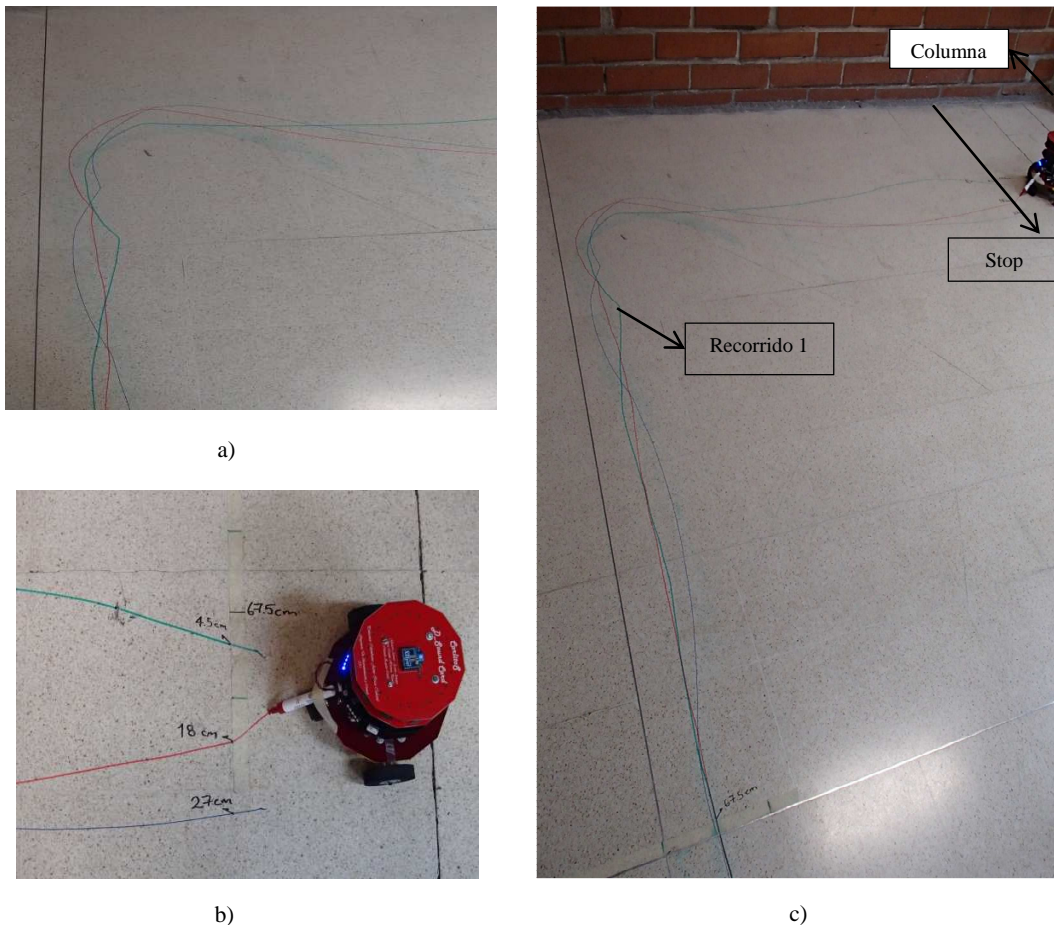


Figura 6.9. a) Evasión esquina izquierda, b) Error de llegada, c) Seguimiento de pared izquierda-evasión de obstáculos

La integración del seguimiento de pared izquierda con la evasión de obstáculos también describió navegaciones exitosas. Se observa que la navegación es más oscilatoria con respecto a los resultados anteriores, pero son aceptables debido a que existe un cumplimiento del objetivo de navegación que es activación de los dos comportamientos mediante el esquema de subsunción. Para una de las pruebas vistas, se presentan a continuación los datos adquiridos para el error frontal, error frontal izquierdo, sensor 10, eS10 y deS4. La figura 6.10 presenta la telemetría de la prueba de seguimiento de pared izquierda con evasión de obstáculos.

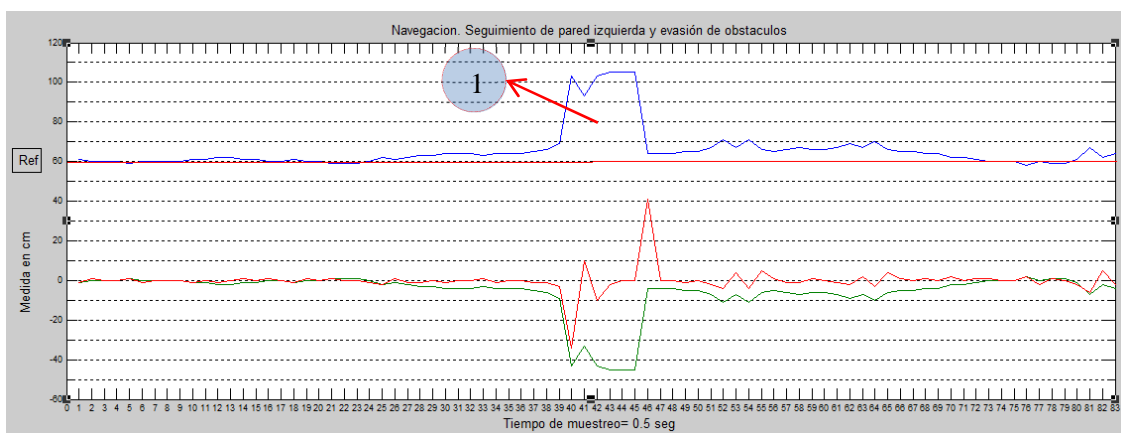


Figura 6.10. Adquisición de datos - Seguimiento de pared Izquierda y evasión de obstáculos.

Se observa el comportamiento esperado para esta prueba de navegación donde la etiqueta 1 indica el avistamiento del vértice izquierdo por parte del sensor S10 en la evasión de la esquina izquierda y se observa también las medidas contra la pared derecha manteniendo el set-point.

La figura 6.11 muestra la detección de obtaculos frontales ubicados a la izquierda del robot.

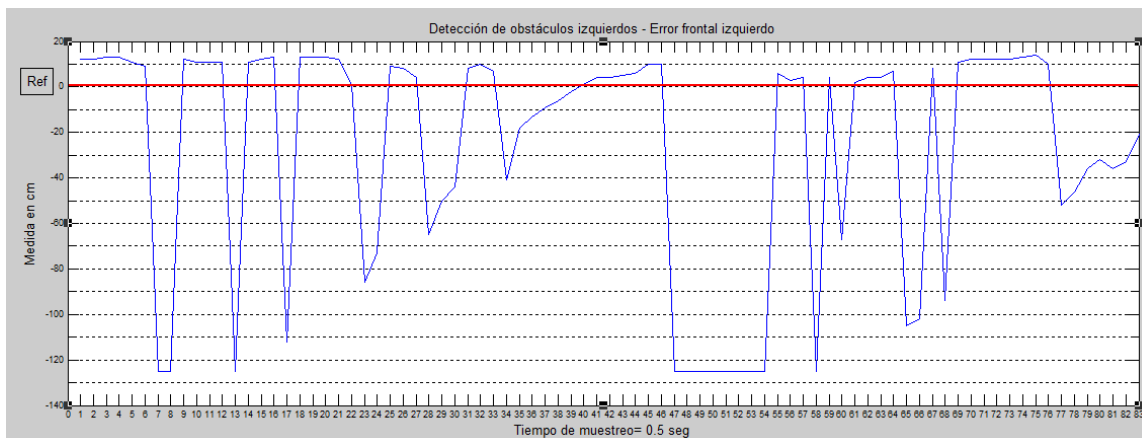


Figura 6.11. Error frontal izquierdo.

Al existir un comportamiento más oscilatorio en el seguimiento de la pared izquierda, los cambios de orientación del robot al corregir hacia la pared son más reiterativos, por esta razón el robot se ve confrontado hacia la pared con mayor frecuencia haciendo que los sensores frontales tenga mayor continuidad avistamientos de objetos a la izquierda que en realidad es la pared que el robot sigue. Por tales razones la figura 6.11 presenta mayores eventos de detección de objetos a la izquierda del robot. La figura 6.12 muestra el avistamiento de objetos frontales.

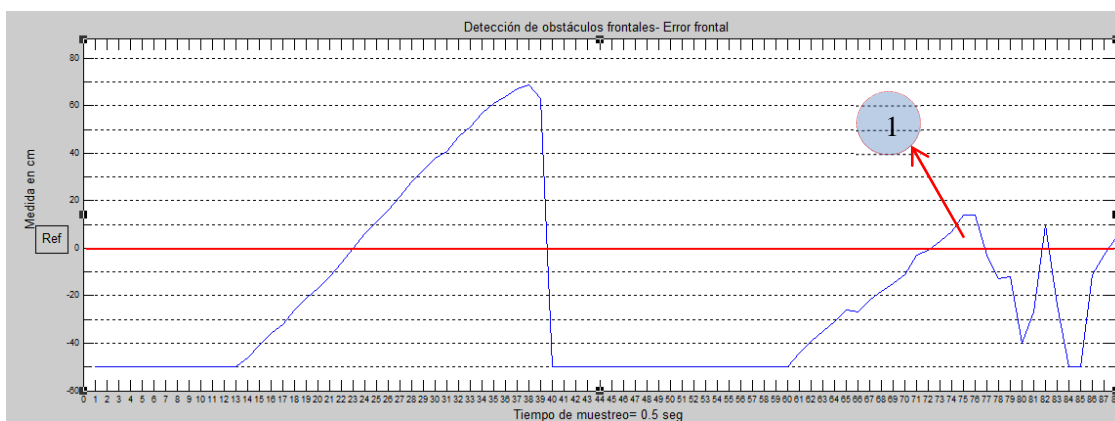


Figura 6.12. Error frontal

En esta grafica vemos en primera instancia la detección de la pared frontal como va aumentando hasta infringir en la zona de detección, lo cual hace que el robot inicie el proceso de evasión en colaboración con los datos de los sensores frontales izquierdos. Se observa otra detección de un objeto frontal en 1, este avistamiento

se asocia a la columna que el robot ve al terminar de recorrer la arena de prueba (ver figura 6.9 - c))

6.1.3. Conclusiones del uso del esquema de subsunción.

En general el uso del esquema de subsunción diseñado para el robot *Carlitos* arrojó buenos resultados de navegación donde claramente se pudo presenciar la dinámica de activación de cada comportamiento según su prioridad.

Si bien con el esquema diseñado se obtuvieron buenos resultados es importante mencionar que al constituirse cada comportamiento en este esquema como una máquina de estado finito (Paolo Pirjanian, 1999) hay que tener en consideración y analizar con detalle los estados en que los efectores quedan cuando existe una transición de un comportamiento a otro.

Para el caso de los seguimientos de paredes donde se controla un solo motor y el otro es constante y viceversa, al pasar del comportamiento de evasión de obstáculos el cual tiene la capacidad de manipular ambos motores al seguimiento de pared, el motor que no es controlado debe ser llevado a las condiciones iniciales debido a las alteraciones que tuvo en el estado anterior, lo que implica una compensación por software al PWM del motor que no se controla en los seguimientos de paredes.

6.2. CONFRONTACIÓN DE LOS ESQUEMAS DE CONTROL

En esta sesión se presentan las pruebas de la evaluación del esquema diseñado en esta investigación versus el esquema de coordinación implementado anteriormente.

La arena de evaluación fue igual que la anterior en la cual el robot describirá de forma experimental la navegación que se ha ido trabajando a lo largo de esta experimentación.

En la figura 6.13 se presenta las navegaciones descritas por el robot utilizando el esquema de subsunción y el esquema de fusión por ponderación contextual dinámica difusa basada en regiones (FPCDBR).

Como actividad de ejecución se escogió el seguimiento de pared derecha y evasión de obstáculos para los dos esquemas como demostración de las navegaciones realizadas. Estas fueron trazadas por el método experimental utilizado para las demás navegaciones documentadas en este trabajo. Inicialmente se realizó la navegación por medio del esquema de subsunción el cual esta descrito por la línea verde en la figura 6.13, posteriormente se reposicionó el robot en la marca de *start* y fue activado el robot para que ejecutara la navegación con FPCDBR.

Estas navegaciones fueron sobrepuestas con el fin de evaluar su desempeño a lo largo de la arena de trabajo.

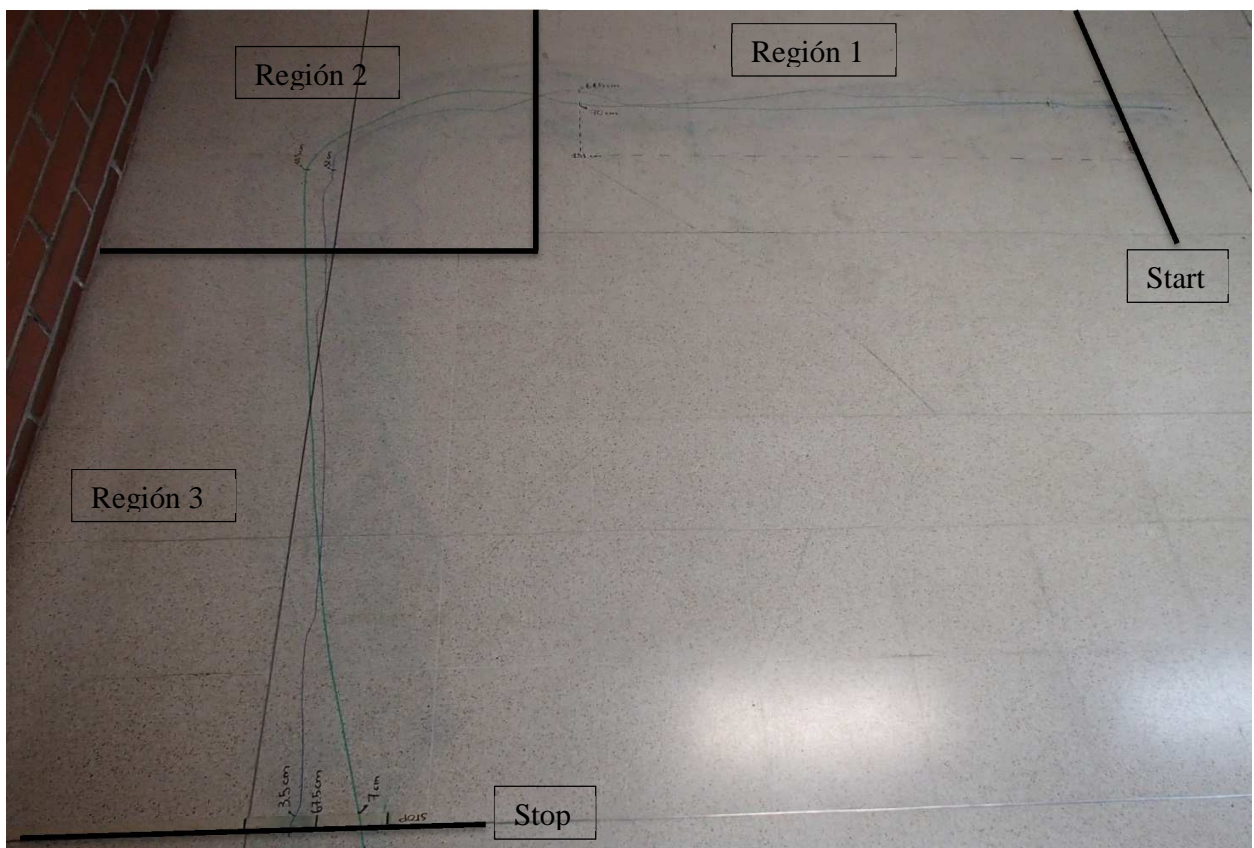


Figura 6.13. Confrontación de esquemas de coordinación de comportamientos.

A continuación se realizará el análisis basado en las regiones de navegación descritas en la figura 6.4 a).

6.2.1. Seguimiento de pared derecha región uno

En este tramo de navegación se evaluó el desempeño de cada uno de los esquemas de coordinación para el seguimiento de la pared derecha. En la figura 6.14 se presenta un zoom de la región analizada.

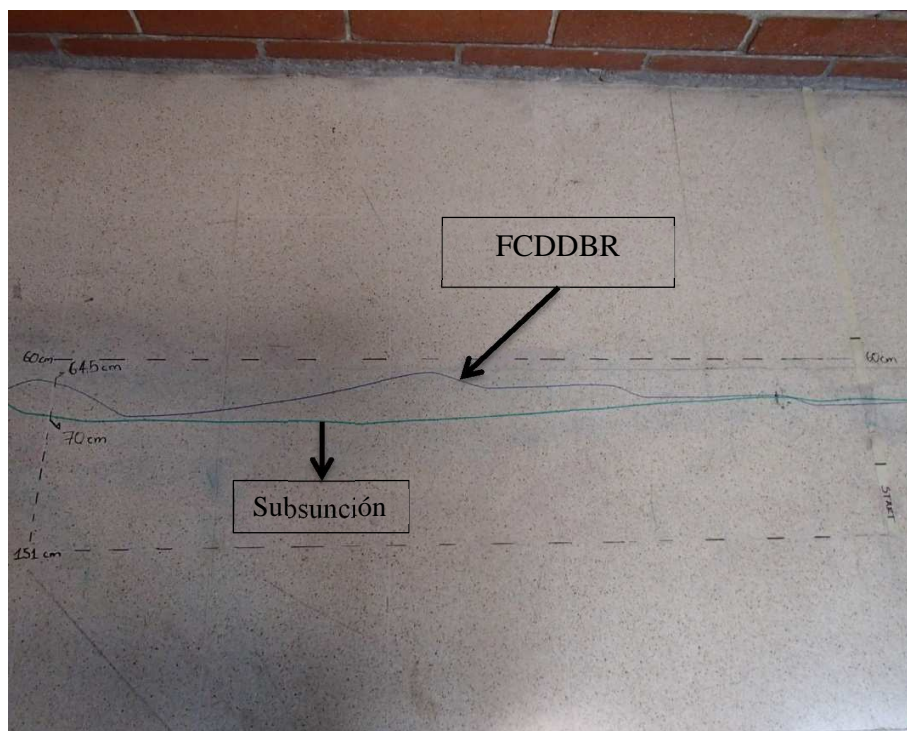


Figura 6.14. Región 1: Seguimiento de pared derecha.

Podemos observar entonces lo siguiente:

- El esquema de subsunción tiende a alejarse de la pared y luego trata de retornar lentamente al *set-point* en búsqueda de disminuir el error contra la pared. Esta

navegación es parecida a la vista en las pruebas de comportamientos independiente debido a que la subsunción inicialmente da prioridad a este comportamiento de seguimiento de pared. Por otro el esquema FPCDBR fue más sensible a los cambios en las regiones lo cual se manifiesta en las oscilaciones descritas.

- Si bien para esta región el esquema FPCDBR fue más oscilante que la subsunción, éste realizó las correcciones siempre llevándolo a la pared y no alejándolo como lo hace la subsunción.
- Se observa principalmente que al final del tramo el esquema FPCDBR tomó la medida de evasión más temprana que la subsunción, esto ayudó considerablemente a que el robot no ingresara mucho hacia la pared en la siguiente región y por ende una evasión más suave.

6.2.2. Evasión de obstáculo región 2

La figura 6.15 muestra la región dos donde se analiza la evasión de obstáculos.

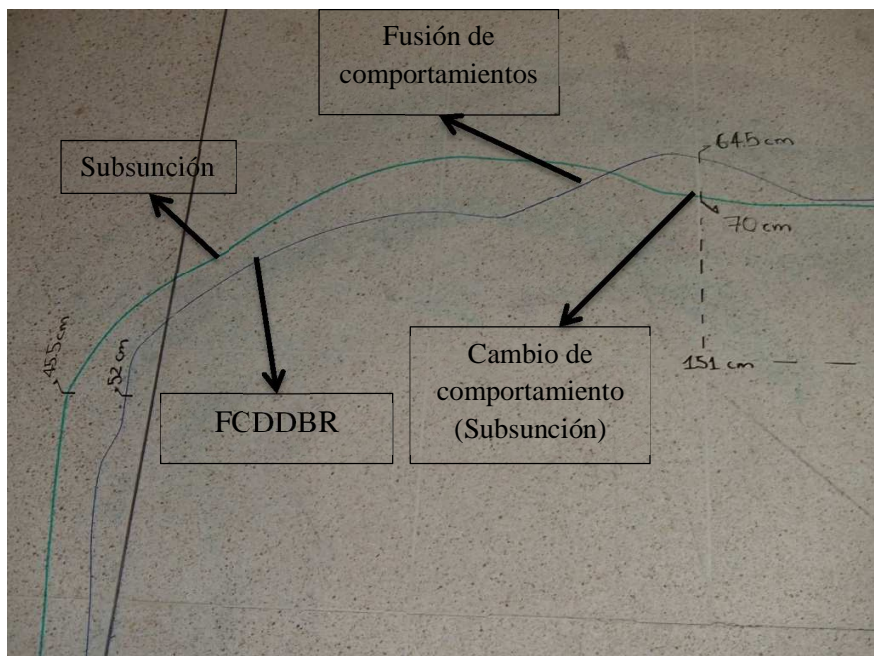


Figura 6.15. Región 2: Evasión de obstáculo.

Gracias a la fusión de las recomendaciones se observa que el diseño de FPCDBR mejora notablemente respecto a la subsunción. El éxito de que el método propuesto realice la corrección de evasión con menos error y de forma más suave es debido a la cooperación que existe en el paso de un comportamiento a otro y la ponderación dinámica asociada al entorno actual. En la subsunción se observa que hasta que no se cumplan las condiciones de evasión no se subsume el seguimiento de pared lo que adentra al robot en cambios bruscos.

6.2.3. Seguimiento de pared derecha región 3

En la figura 6.16 se presenta en tramo final de la prueba de evaluación donde se observa el cambio de la evasión de obstáculos, al seguimiento de pared derecha.

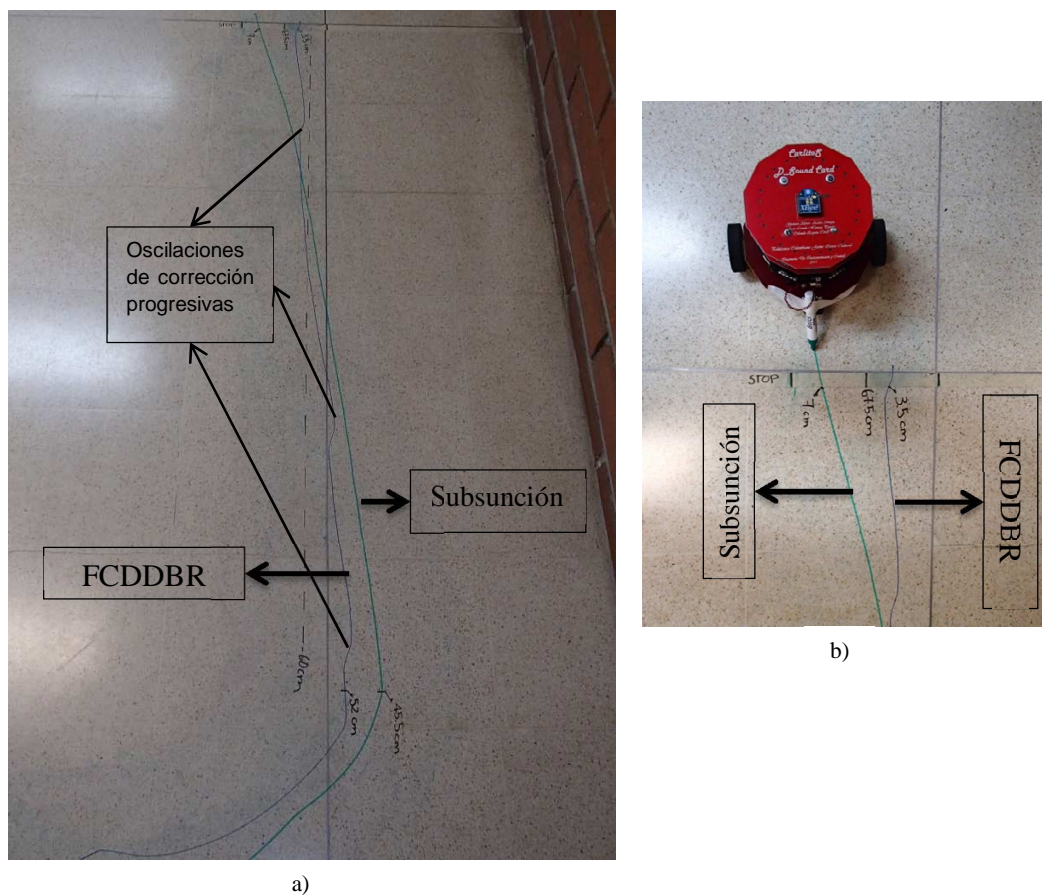


Figura 6.16. a) Región 3: Seguimiento de pared derecha; b) Error de llegada.

Se analiza que el esquema de subsunción al terminar el estado de evasión de obstáculo retoma el seguimiento de pared el cual intenta sacar el robot apresuradamente debido a la cercanía que existe ente el robot y la pared.

Estas correcciones bruscas llevan al sistema a terminar el recorrido con un error de llegada mayor. Por otra parte se observa que el esquema de FPCDBR termina la evasión la esquina derecha y comienza a realizar correcciones progresivas en búsqueda del *set-point*. Estas correcciones progresivas hacen que el sistema no se pierda y por ende el error de llegada disminuya.

En la figura 6.17 se confrontan las bases de datos recolectadas en las navegaciones analizadas arriba.

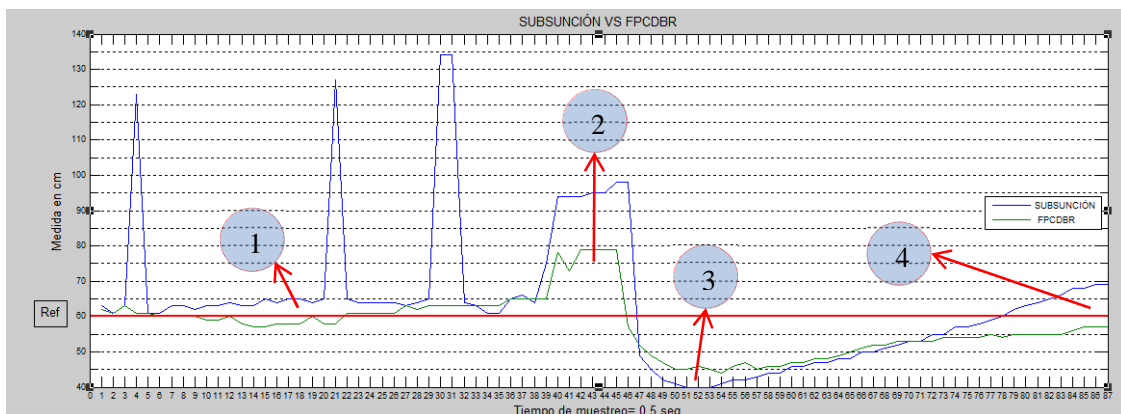


Figura 6.17. SUBSUNCIÓN VS FPCDBR

En la gráfica de confrontación de las navegaciones por medio de los datos adquiridos podemos corroborar todas nuestras hipótesis planteadas para las pruebas de navegación donde una fusión por ponderación contextual dinámica difusa basada en regiones mejoraría la navegación del robot móvil (FPCDBR).

Analizando las etiquetas puestas en la figura podemos analizar lo siguiente

1. La navegación en FPCDBR mejora manteniendo el *set-point* con mayor frecuencia.

2. La evasión de esquinas en FPCDBR es más suave y con menos error.
3. La fusión entre las recomendaciones de los comportamientos en FPCDBR presento estados transitorios más sutiles.
4. El error de llegada disminuye gracias a la fusión de las recomendaciones de los comportamientos en la navegación.

6.3.CONCLUSIONES DE CAPÍTULO

- Al confrontar los dos esquemas tratados en este capítulo se puede apreciar sus ventajas y desventajas. En cuanto al esquema de subsunción podemos aseverar que aunque en algunos casos los resultados son satisfactorios y se debe tener cuidado con los estados de cada comportamiento y los elementos finales de control puesto que en algunos casos el paso de un comportamiento a otro puede generar acciones erróneas de control que pueden hacer que el sistema se pierda y se desestabilice por completo.
- Los resultados de navegación fueron exitosos para el diseño implementado, se observaron mejorías notorias frente al esquema con el que fue comparado, se obtuvo una buena coordinación de los comportamientos diseñados y hubo concordancia entre las pruebas experimentales y los datos adquirido por el aplicativo de adquisición, lo cual indica que se obtuvo una buena apropiación tanto de los componentes de hardware y software.
- Fue fundamental el conocimiento de los expertos para ser materializados en el diseño y también fue clave el conocimiento al máximo del sistema robótico.
- Para el sistema de control diseñado fue indispensable mantener una estrategia que permitiera realizar de manera inmediata ajustes en las labores de rediseño tanto en los comportamientos como en el sistema de control global.

Conclusiones.

- Después de todo el trabajo en conjunto se evidencia que se realizó el cumplimiento de todos los objetivos y se manifiesta de manera positiva el cumplimiento de la hipótesis planteada en esta investigación donde podemos aseverar que un esquema de control basado en comportamientos administra de manera eficiente las conductas del robot, evitando conflictos entre ellos por el controlar los efectores.
- La implementación del esquema FPCDBC mejoró la navegación del sistema robótico de medianas prestaciones. Este esquema permitió que el robot describiera trayectorias suaves de navegación con una efectiva administración de comportamientos y la robustez necesaria para manejar los problemas de incertidumbre sensorial que se presentaron en las pruebas de diseño y validación.
- Durante esta investigación fue vital el conocimiento y apropiación tanto de los componentes de hardware y software para poder suplir los problemas que fueron surgiendo paulatinamente. Hay que resaltar que la sintonización de los parámetros de los comportamientos fue transversal a todo el proceso en busca una navegación más segura y con mejores respuestas.
- La integración de los comportamiento diseñados al esquema FPCDBC fue exitosa ayudando al robot *Carlitos* a mejorar su autonomía en la navegación brindándole de manera efectiva conocimiento del entorno el línea para que no sufriera ningún tipo de colisión contra objetos.
- Las estructuras de control difusas basadas en comandos de fusión cooperativos han sido trabajadas ampliamente en múltiples partes del mundo y los resultados que han obtenido las comunidades científicas que trabaja en robótica móvil basada en el panorama reactivo han sido muy satisfactorios. Ésta investigación presenta un aporte significativo a nivel local en el área de la robótica móvil y que

también se constituye como una buena contribución los mecanismos de coordinación ya existentes.

Trabajo a futuro

- Evaluar otros esquemas de control de navegación en el robot móvil utilizando otra clase de técnicas de inteligencia artificial tales como las redes neuronales, sistemas neurodifusos, sistemas difusos basados en algoritmos genéticos.
- Utilizar técnicas de optimización como los algoritmos genéticos o computación evolutiva para implementar estrategias de *autotuning* de los parámetros de los controladores difusos asociados a los comportamientos.
- Realizar estudios de localización del robot por medio de la odometría y mejorar la navegación del autómata.

Bibliografía

- Abreu, A., & Correia, L. (1999). Fuzzy Behaviors and Behavior Arbitration in Autonomous Vehicles. In P. Barahona & J. J. Alferes (Eds.), *Progress in Artificial Intelligence* (pp. 237–251). Springer Berlin Heidelberg. Retrieved from http://link.springer.com.ezproxy.unal.edu.co/chapter/10.1007/3-540-48159-1_17
- Acosta Amaya, G. A. (2010). *Ambiente multi-agente robótico para la navegación colaborativa en escenarios estructurados / Multi-agent robotic environment for collaborative navigation in structured scenarios* (masters). Universidad Nacional de Colombia. Retrieved from <http://www.bdigital.unal.edu.co/2533/>
- Adriansyah, A., & H. M. Amin, S. (2005). Knowledge base tuning using genetic algorithm for fuzzy behavior-based autonomous mobile robot (p. ICMT–153). Presented at the Proceeding of the 9th International Conference on Mechatronics Technology, Kuala Lumpur. Retrieved from <http://eprints.utm.my/1841/>
- Amin, S. H. M., & Adriansyah, A. (2006). Particle Swarm Fuzzy Controller for Behavior-based Mobile Robot. In *9th International Conference on Control, Automation, Robotics and Vision, 2006. ICARCV '06* (pp. 1–6). doi:10.1109/ICARCV.2006.345293
- Arkin, R.C. (1987). Motor schema based navigation for a mobile robot: An approach to programming by behavior. In *1987 IEEE International Conference on Robotics and Automation. Proceedings* (Vol. 4, pp. 264–271). doi:10.1109/ROBOT.1987.1088037
- Arkin, Ronald C. (1998). *Behavior-based Robotics*. MIT Press.

- Benet, G., Blanes, J. F., Simon, J. E., Vila, J., & Crespo, A. (1998). "Desarrollo de sistemas de robótica móvil en entornos de trabajo industriales."
- Bräunl, T. (2008). *Embedded robotics: mobile robot design and applications with embedded systems ; with 32 tables*. Berlin; Heidelberg: Springer.
- Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1), 14–23. doi:10.1109/JRA.1986.1087032
- Dongshu, W., Yusheng, Z., & Wenjie, S. (2011). Behavior-based hierarchical fuzzy control for mobile robot navigation in dynamic environment. In *Control and Decision Conference (CCDC), 2011 Chinese* (pp. 2419–2424). doi:10.1109/CCDC.2011.5968614
- Farooq, U., Khalid, A., Amar, M., Habiba, A., Shafique, S., & Noor, R. (2010). Design and low cost implementation of a fuzzy logic controller for wall following behavior of a mobile robot. In *2010 2nd International Conference on Signal Processing Systems (ICSPS)* (Vol. 2, pp. V2–740–V2–746). doi:10.1109/ICSPS.2010.5555781
- Freescall. (2013). MCF51QE: Flexis 32-bit ColdFire V1 Microcontroller. Retrieved November 2, 2013, from http://www.freescall.com/webapp/sps/site/prod_summary.jsp?code=MCF51QE&nodelid=018rH3YTLC00M924B2
- Hagras, H., Callaghan, V., Colley, M., & Carr-West, M. (1999). A behavior based hierarchical fuzzy control architecture for agricultural autonomous mobile robots. In *Int. Conf. Computational Intelligence for Modeling, Control, and Automation*. Vienna, Austria.
- Incon - Produtos Eletrônicos. (2013). Retrieved November 2, 2013, from <http://www.incon.com.br/index.php?sec=produtos&funcao=detalhes&id=88>
- Jiménez, J., Vallejo, M., & G., J. F. O. (2007). Metodología para el Análisis y Diseño de Sistemas Multi-Agente Robóticos: MAD-Smart. *RASI*, 4(2), 61–70.

- KARSULOVIC, J. T., BOZO, A., ARAYA, J., VARGAS, Y., GAETE, V., & TEJER, B. (2008). APLICACIÓN DE ONDAS SÓNICAS Y ULTRASÓNICAS COMO MEDIO DE CONTROL DEL ATAQUE DE TERMITAS SUBTERRÁNEAS RETICULITERMES FLAVIPES (KOLLAR) EN MADERA DE PINO RADIATA. *Maderas. Ciencia y Tecnología*, 10(2), 99–111. doi:10.4067/S0718-221X2008000200003
- Liu, G., Yao, M., Zhang, L., & Zhang, C. (2011). Fuzzy Controller for Obstacle Avoidance in Electric Wheelchair with Ultrasonic Sensors. In *2011 International Symposium on Computer Science and Society (ISCCS)* (pp. 71–74). doi:10.1109/ISCCS.2011.27
- Liu, Z., & Kubota, N. (2007). Hybrid Learning Approach based on Multi-Objective Behavior Coordination for Multiple Robots. In *International Conference on Mechatronics and Automation, 2007. ICMA 2007* (pp. 204–209). doi:10.1109/ICMA.2007.4303541
- Llata, J. R., Sarabia, E. G., & Oria, J. P. (2002). Three-Dimensional Robotic Vision Using Ultrasonic Sensors. *Journal of Intelligent and Robotic Systems*, 33(3), 267–284. doi:10.1023/A:1015084304191
- MacKenzie, D. C., Arkin, R., & Cameron, J. M. (1997). Multiagent Mission Specification and Execution. *Autonomous Robots*, 4(1), 29–52. doi:10.1023/A:1008807102993
- Navarro, D., Ríos, L. H., & Parra, H. (2004). SENSORES DE ULTRASONIDO USADOS EN ROBÓTICA MÓVIL PARA LA MEDICIÓN DE DISTANCIAS, 25(Scientia et Technica Año X).
- Navarro, Danilo, Gilabert, G. B., Rios, L. H., & L, M. B. (2007). Mejoras de la localización odométrica de un robot diferencial mediante la corrección de errores sistemáticos. *Scientia Et Technica*, XIII(37), 37–42.
- Oria, J., & Gonzalez, A. M. G. (1993). Object recognition using ultrasonic sensors in robotic applications. In , *International Conference on Industrial Electronics, Control,*

- and Instrumentation, 1993. Proceedings of the IECON '93* (pp. 1927–1931 vol.3).
doi:10.1109/IECON.1993.339369
- Peng, J., & Yumei, H. (2009). Behavior-Based Avoiding Barriers System of Mobile Robot.
In *WRI World Congress on Software Engineering, 2009. WCSE '09* (Vol. 3, pp. 106–112). doi:10.1109/WCSE.2009.307
- Pirjanian, P., & Mataric, M. (2000). Multi-robot target acquisition using multiple objective behavior coordination. In *IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA '00* (Vol. 3, pp. 2696–2702 vol.3).
doi:10.1109/ROBOT.2000.846435
- Pirjanian, Paolo. (1999). *Behavior Coordination Mechanisms - State-of-the-art*.
- Pololu. (2013). Pololu - Metal Gearmotor. Retrieved November 2, 2013, from
/product/2288
- Qing-yong, B., Shun-ming, L., Wei-yan, S., & Mu-jin, A. (2009). A Fuzzy Behavior-Based Architecture for Mobile Robot Navigation in Unknown Environments. In *International Conference on Artificial Intelligence and Computational Intelligence, 2009. AICI '09* (Vol. 2, pp. 257–261). doi:10.1109/AICI.2009.125
- Rosenblatt, J. (1995). DAMN: A Distributed Architecture For Mobile Navigation - Thesis Summary. In *Journal of Experimental and Theoretical Artificial Intelligence* (pp. 339–360). AAAI Press.
- Safiotti, A. (1997). Fuzzy logic in autonomous robotics: behavior coordination. In ,
Proceedings of the Sixth IEEE International Conference on Fuzzy Systems, 1997
(Vol. 1, pp. 573–578 vol.1). doi:10.1109/FUZZY.1997.616430
- Sarabia, E. G., Llata, J. R., Arce, J., & Oria, J. P. (1998). Shape recognition and orientation detection for industrial applications using ultrasonic sensors. In , *IEEE International Joint Symposia on Intelligence and Systems, 1998. Proceedings* (pp. 301–308). doi:10.1109/IJSIS.1998.685464

- Shim, B.-K., Cho, Y.-K., Lee, W.-S., & Han, S.-H. (2011). A robust control of mobile robot based on sonar sensors. In *2011 11th International Conference on Control, Automation and Systems (ICCAS)* (pp. 1106–1107).
- Song, K.-T., & Lin, J.-Y. (2006). Behavior Fusion of Robot Navigation Using a Fuzzy Neural Network. In *IEEE International Conference on Systems, Man and Cybernetics, 2006. SMC '06* (Vol. 6, pp. 4910–4915).
doi:10.1109/ICSMC.2006.385083
- Super Robotica. (2013). Super Robotica. *www.superrobotica.com*. Retrieved November 2, 2013, from <http://www.superrobotica.com/s320122.htm>
- Tang, H., Song, A., & Zhang, X. (2006). Hybrid Behavior Coordination Mechanism for Navigation of Reconnaissance Robot. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1773–1778).
doi:10.1109/IROS.2006.282217
- Thongchai, S., Suksakulchai, S., Wilkes, D. M., & Sarkar, N. (2000). Sonar behavior-based fuzzy control for a mobile robot. In *2000 IEEE International Conference on Systems, Man, and Cybernetics* (Vol. 5, pp. 3532–3537 vol.5).
doi:10.1109/ICSMC.2000.886556
- Thrun, S., Bennewitz, M., Burgard, W., Cremers, A. B., Dellaert, F., Fox, D., ... Schulz, D. (1999). MINERVA: a second-generation museum tour-guide robot. In *1999 IEEE International Conference on Robotics and Automation, 1999. Proceedings* (Vol. 3, pp. 1999–2005 vol.3). doi:10.1109/ROBOT.1999.770401
- Wicaksono, H., PKASKREA, A., & Mitsuji, S. (2009). APPLICATION OF FUZZY BEHAVIOR COORDINATION AND Q LEARNING IN ROBOT NAVIGATION. *Thesis/Dissertation*. Retrieved from http://fportfolio.petra.ac.id/user_files/04-004/ICTS%202009%20Handy%20Wicaksono.pdf

- Zapata, O., & Herrera, C. (2011). *PLATAFORMA ROBÓTICA PARA EL MAPEO Y NAVEGACIÓN EN ENTORNOS INTERIORES* (Trabajo de grado) (p. 135).
Medellín: POLITÉCNICO COLOMBIANO JAIME ISAZA CADAVID FALCULTAD DE INGENIERÍAS.
- Zhang, H., Liu, S., & Yang, S. X. (2006). A Hybrid Robot Navigation Approach Based on Partial Planning and Emotion-Based Behavior Coordination. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1183–1188).
doi:10.1109/IROS.2006.281851

Anexos A. Diagramas de Flujo de programas y subrutinas

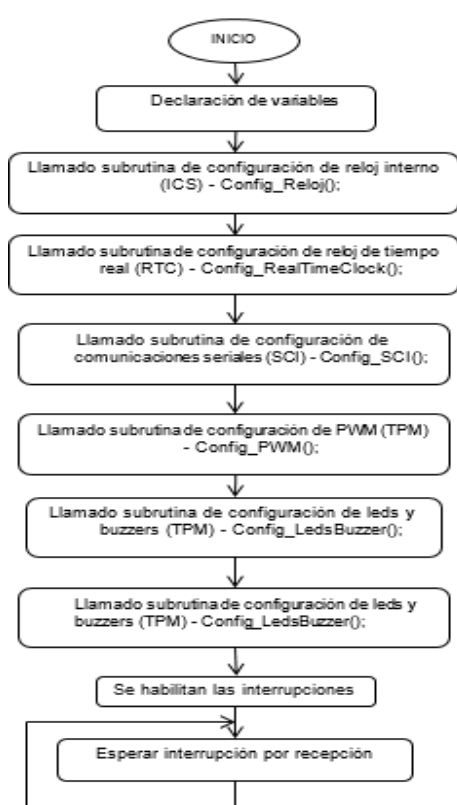


Figura A1. Diagrama de flujo programa principal.

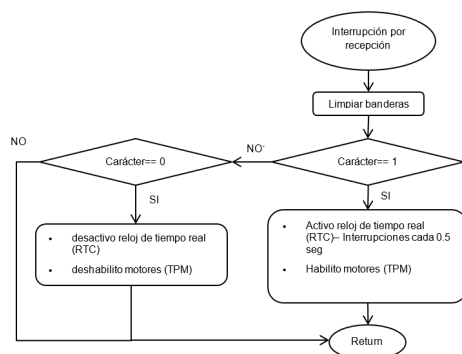


Figura A2. Rutina de atención a interrupción por recepción de datos del puerto serial en el robot *Carlitos*.

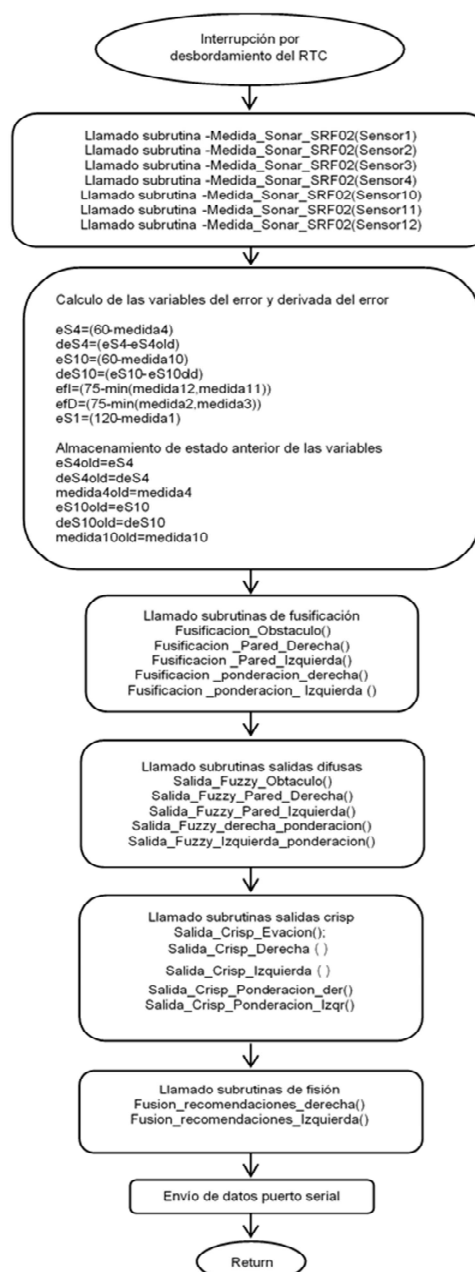


Figura A3. Rutina de atención a interrupción por desbordamiento del reloj de tiempo real.

Anexos B. Interfaz de adquisición de datos

TOMA DE LECTURAS

TELEMETRÍA DEL ROBOT CARLITOS

TELEMETRÍA

	S4	eS4	dS4
SENSOR COSTADO DERECHO	0	0	0

	minI	S1	minD	efl	efD
SENSORES FRONTALES	0	0	0	0	0

	S10	eS10	dS10
SENSOR COSTADO IZQUIERDO	0	0	0

PESOS

PESOS SEGUIMIENTO DE PARED	0
PESOS EVASIÓN DE OBSTÁCULOS	0

BOTONES DE COMANDO

START STOP

Guardar Datos

Figura B 1. Interfaz diseñada para adquirir los datos de los sensores.

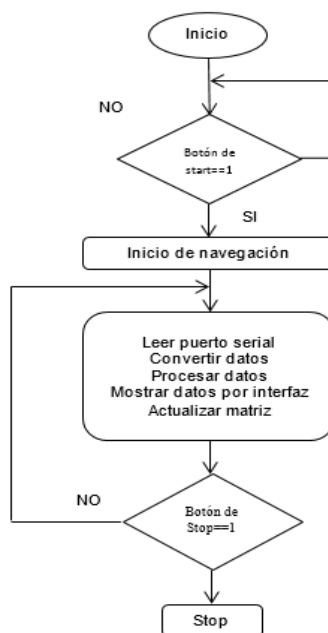


Figura B 2. Diagrama de flujo del software de adquisición de datos.